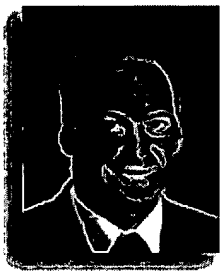
DBMS, August 1998

DBMS online



Server Side

Martin Rennhackkamp

IBM's Intelligent Family

Decision making driven by IBM's Business Intelligence Initiative.

In its press releases, IBM describes business intelligence as the gathering, management, and analysis of data for the purpose of turning it into useful information distributed throughout an enterprise and used to improve strategic decision making.

When we look at IBM's business intelligence initiative in more detail, we see that it consists of a wide portfolio of elements integrated from many of IBM's diverse divisions to offer, in the company's words, a truly complete, end-to-end solution to meet complex business challenges. As a result, the business intelligence initiative consists of solutions and applications to address specific industries or related industries, such as banking or insurance; data analysis products to mine and interpret structured, textual, and Web data; consulting, services, and sales by IBM's business intelligence specialists to provide integration, implementation, and support for a customized business intelligence solution; specialized partnership, marketing, and development agreements with ETI, Cognos, Vality, Arbor, Per-Se, and Business Objects for tools and applications to access, extract, cleanse, and analyze data; data management software for the deployment and management of small to very large data warehouses; advanced mathematical algorithms developed by IBM Research for data analysis; database products, such as DB2 Universal Database (UDB) for creating, accessing, and managing traditional and multimedia data types; and hardware platforms, such as S/390, RS/6000, AS/400, and Netfinity, that offer scalable performance for the computation needs of advanced business intelligence.

Of this wide range of elements, I will focus on the server components - the data mining, data analysis, and data warehousing products - in the context of the wider business intelligence initiative and related to IBM's DB2 UDB database management systems.

Data Mining

IBM's business intelligence initiative includes a number of products in the data mining field, including

the well-known Intelligent Miner family and more obscure products such as Fast Lookup Algorithm for Structural Homology (FLASH) and Teiresias. The Intelligent Miner family of products is aimed at extracting previously unknown, comprehensible information from any data source. It consists of the Intelligent Miner for Text and version 2 of the Intelligent Miner for Data. IBM also markets a set of applications, called the IBM Discovery Series, that sits on top of the IBM Intelligent Miner for Data, solving specific business problems through data mining.

The Intelligent Miner for Text has three major components: a search engine called TextMiner, Web access tools including a Web search engine called NetQuestion and a Web crawler, and text analysis tools. Figure 1 shows Intelligent Miner for Text's architecture. You can use this combination of tools to analyze documents such as word processing documents, online news articles and email messages to group and prioritize information contained in the text data. It can discover in which language a document is written, and it can extract names, multiword terms, abbreviations, and other vocabulary such as dates, figures, and amounts. It extracts patterns, organizes documents by subject, finds predominant themes, and searches for relevant documents.

Many of these tools are information extractors that enrich documents with information concerning their contents because the first step in text mining is to extract key features from texts to act as "handles" in further processing. The information retrieval component uses hash indexes built offline to perform Boolean or relevance ranking queries to select text documents. The TextMiner search engine provides full-text search and indexing of documents written in 16 languages, including double-byte languages such as Japanese, Chinese, and Korean, stored in many different file formats, using natural language, free-text, Boolean, fuzzy, phonetic and hybrid search conditions. The patented hybrid queries, for example, combine free-text and Boolean queries to overcome the problems of pure free-text queries. A hybrid query is a free-text query that restricts the result set to the documents that also match the Boolean part of the query. This allows for negative specifications in free-text queries that are not supported by pure free-text systems. The NetQuestion Web search engine, on the other hand, although it uses the same techniques as TextMiner, is streamlined for the types of information typically found on Web pages. You can use it for Boolean queries and phrase and proximity searches, as well as for front, middle, and end masking using wild cards.

Intelligent Miner for Text provides extensive functionality. Although computers do not easily process unstructured text, it is becoming the predominant datatype stored online. This is evident from the ever-growing number of news groups and email messages, not to mention Web pages and word processing documents. There is an amazing amount of useful information contained in such unstructured data.

The Intelligent Miner for Data searches for hidden information, associations, or patterns. It clusters data records based on similar values, using a voting technique called Condorset. It segments data using neural clustering - a technique that employs a type of neural network called the Kohonen feature map that clusters together similar data records and defines the typical attributes of an item that falls in a given cluster or segment. It discovers associations, sequential patterns, and similar time sequences and creates predictive or classification models of the data. It performs deviation detection by relying heavily on statistical analysis and visualization. The visualization techniques are useful for detecting deviations that hold for a rather small subset of the data, while it uses statistics to measure their significance.

You can use the Intelligent Miner for Data to analyze data stored in traditional files, relational databases, data warehouses, and data marts. Version 2.1 has a new Java-based graphical user interface with hover help, pop-up context menus, SmartGuides, optional hiding of advanced features, user preference settings, a progress indicator, graphical representation of the mining base and mining objects, and a graphical construction mechanism for composite objects. It uses new and enhanced statistical functions, algorithms, and optimized mining techniques, such as factor analysis, linear regression, principal

component analysis, univariate curve fitting, univariate statistics, bivariate statistics, and logistic regression. It contains a new neural net implementation of the value prediction method, and its mining techniques have been optimized to handle outliers, missing values, and lift.

Version 2.1 runs on more platforms, exploits the DB2 UDB functionality, and is more scalable than the previous release. It can mine data in flat files or other databases accessible through DataJoiner, such as Sybase or Oracle. You can use its high-speed extract facility to import data into DB2 UDB from Oracle, Sybase, or DB2 for OS/390. On DB2 UDB, it uses parallelized versions of the mining algorithms for large-scale mining runs. The Intelligent Miner for Data supports English, French, Hungarian, Italian, Japanese, Korean, Portuguese, Russian, Spanish, and traditional Chinese data. In addition to providing a published API as a client interface, it also provides a server API.

FLASH is an advanced pattern-matching algorithm designed to identify similar, but not identical, data. It is useful in business applications such as insurance and finance, DNA and genetic matching research, biometrics, rational drug design and other data-intensive work.

Teiresias is a pattern-discovery algorithm used in a joint research project by IBM and Monsanto to reduce the research and development cycle for life sciences products. You can apply it to any database to discover previously unknown patterns quickly.

Similar to text mining, data mining discovers hidden patterns in the data - patterns that, when analyzed appropriately, can contribute to the organization's prosperity. Although you may think that structured data, as stored in a database, may have all the relationships, meanings, and constraints identified, you can gain a lot of knowledge by analyzing that same data with data mining tools. Patterns hidden within the data or within relationships among the data can, for example, guide decision makers to different marketing models.

Data Analysis

The DB2 OLAP Server integrates the OLAP engine and APIs of Arbor Software's Essbase analytical processing engine with IBM's DB2 UDB. All the design, manipulation, calculation, and analysis functions of the Essbase server are available in the DB2 OLAP server. The DB2 OLAP Server stores and manages the data in a DB2 UDB database using a star schema structure. You can populate the star schema with data processed by the Essbase calculation engine to improve query performance. You can access the data from clients supporting the Essbase client API or from any other clients using standard SQL. The API supports Visual Basic, C, C++, and other application development environments and works with Windows 95 or NT, Unix, and OS/2.

There are also a number of tools you can use with the DB2 OLAP Server to extend its functionality further. The Web Gateway provides the standard OLAP functions from standard Web browsers. The Extended Spreadsheet Toolkit includes more than 20 macros and Visual Basic functions to integrate custom Lotus 1-2-3 or Microsoft Excel applications with the DB2 OLAP Server. SQL Drill-Through provides links between summary data in the DB2 OLAP Server and detailed data in relational databases. The Partitioning Option makes it possible to design and manage multidimensional databases (cubes or star schemas) that span OLAP applications or servers. The Adjustment Module integrates secure, auditable controls for adjustments into a comprehensive reporting, analysis, and planning environment. The Currency Conversion converts financial data using different currency exchange rates. The SQL Interface provides direct access to more than 20 PC and SQL relational databases, including Oracle, Sybase, Informix, Microsoft SQL Server, and other middleware packages. Objects are a set of open, ActiveX, OLAP-aware objects through which users can develop OLAP applications on Windows 95 and

NT clients with minimal programming.

Data Warehousing

Visual Warehouse is IBM's data warehouse and data mart solution targeted at the entire range of data warehouses - from departmental data marts up to the enterprise data warehouse. Visual Warehouse is not a single system, but a family of integrated tools. It consists of three components that cover the various steps in building, managing, and analyzing data warehouses and data marts: Visual Warehouse Desktop, Operations, and Agents. You use Visual Warehouse Desktop to define metadata, such as sources, targets, and mapping transformations. Operators use Visual Warehouse Operations to manage and monitor the various operational procedures related to data warehouse processing. The Visual Warehouse Agents perform the actual tasks specified by the Desktop and Operations components. You can also extend Visual Warehouse with specialized third-party tools for some of these steps.

Visual Warehouse Desktop provides the facilities to define relationships and mappings between online data and the data warehouse. It can map data from DB2, DB2 UDB, Oracle, Informix, SQL Server, CICS/VSAM, or IMS. Visual Warehouse can interoperate with other systems through metadata interchange facilities based on IBM's metadata tag language or the Metadata Coalition's Metadata Interchange Specification, an industry standard that simplifies interoperability among CASE tools; repositories; analysis tools; and extract, transformation, movement, and loading tools.

Visual Warehouse provides various tools to consolidate, cleanse, restructure, correlate, standardize, and summarize data from multiple source systems through so-called Business Views. The Business Views control how the data is transformed from the source databases into meaningful business information and automatically extracted, transferred, transformed, and refreshed in the data warehouse. The Business Views are graphically specified aggregations, summaries, and derivations of the source data, where multiple data sources are combined in a single view for decision makers. Visual Warehouse can maintain multiple copies of these data sets. You can specify the number of copies it must keep and when older copies must be deleted.

You can also extend Visual Warehouse with various third-party tools. The ETI-Extract tool suite from Evolutionary Technologies International is a loader generator that generates extraction and transformation programs from a visual specification from virtually any data source to any data target in any programming language. The Vality Integrity Data Re-engineering tool from Vality Technology Inc. provides facilities to uncover hidden, undocumented values from legacy systems and correlate information across independent systems to deliver high-quality input data for the data warehouse. Integrity is reputed to be one of the best name and address "scrubbing" tools on the market.

IBM also markets its entire Data Replication Solution as full and differential data warehouse loading tools. This includes DataPropagator Relational Version 5, DataPropagator NonRelational, DataRefresher, DataJoiner, and Infospeed. DataPropagator Relational Version 5, for example, can incrementally capture changed source data for propagation to one or more data warehouses. DataJoiner, in conjunction with Visual Warehouse, provides a single SQL interface to access and join data from a wide variety of non-IBM source databases, such as Oracle, Sybase, and Informix. Infospeed replicates data from S/390s to a wide variety of Unix and NT servers.

In addition, Visual Warehouse provides SQL, incremental updates, and bulk and parallel loading facilities to populate data warehouses.

You can store the data warehouse data in any DB2 database. Visual Warehouse comes bundled with

DB2 UDB, but it can also use DB2 for OS/400 and DB2 for MVS. DB2 UDB is IBM's preferred platform because it is easy to manage through the DB2 Control Center, it is scalable from a single processor to SMP to MPP environments and can thus support hundreds of users and gigabytes to terabytes of data, and it has the facilities to process complex data warehouse queries in parallel. DB2 UDB's query rewrite facility is especially useful for optimizing the poorly structured queries often generated by the popular query and reporting tools. Visual Warehouse can also populate data warehouses implemented using other DBMSs through DataJoiner, IBM's multivendor database middleware that provides access to Oracle, Sybase, and Informix. Through IBM's Cross Platform Attachment, you can store the data from a Unix or Windows NT data warehouse on S/390 storage facilities to reuse its excess storage capacity and take advantage of the S/390 storage management facilities.

The Visual Warehouse Operations components include scheduling and monitoring tools for periodic building and refreshing of the data warehouse through which DBAs can concentrate on managing exceptions rather than worrying about day-to-day operations. It collects status information and statistics about the build processes to enable analysis and tuning that ensures that these maintenance tasks are as efficient as possible. Operators can customize their views to focus on their most important tasks through the new Work-In-Process console.

Visual Warehouse encompasses various tools for accessing and analyzing the data in the data warehouse. It supports a range of data access options including ODBC, JDBC, native DB2 clients, industry-standard SQL, and data analysis through the Arbor's Essbase OLAP API. You can order it with data analysis tools from Business Objects or Cognos as part of the package. It can also provide wide-range information access from the Web or an intranet through all the familiar Web browsers.

The latest release, Visual Warehouse 3.1, runs on AIX, OS/2, and Windows NT. The addition of AIX and OS/2 results in improved performance for data warehouses on those platforms, because data does not need to flow through a Windows NT agent.

Intelligent Mix and Match

It should be obvious from this column that IBM's business intelligence initiative is not a single product, but rather the integration of a wide range of products - some new, some established - from various subject areas, in an attempt to address all the needs of the business users. In other words, the goal is to transform any variety of data source into a single integrated information source that you can use for more intelligent business decisions.

What's encouraging to me is that it is not merely a conglomerate of quasirelated proprietary products, but rather an open solution in which you can use the most applicable or best-of-breed products. For example, replication is not always a viable data warehouse population technique because of the complex transformations you sometimes have to perform. Similarly, an organization doesn't necessarily want to store its warehouse data in DB2 UDB or access it through tools supporting the Essbase API. With this solution, you can use specific best-of-breed products. For example, Essbase, BusinessObjects, and Cognos's Impromptu and PowerPlay are, in my opinion, the most useful and most widely used data analysis tools. Similarly, ETI's ETI-Extract is a very flexible, extendible data warehouse loader generator. It is one of the few tools in which you can include customized transformations written in C++, C, or Visual Basic to perform more than the traditional aggregation and summarization transformations.

Using IBM's business intelligence initiative, an organization can now implement its data warehouse or

data mart solution to improve its decision-making process using an intelligent mix and match of the appropriate products and system components without being forced down a particular avenue that it may not want to follow. The solution is closed enough to ensure proper integration of the data sources into the data warehouse, but open enough to let users pick the appropriate mix and match of tools to satisfy their information requirements.

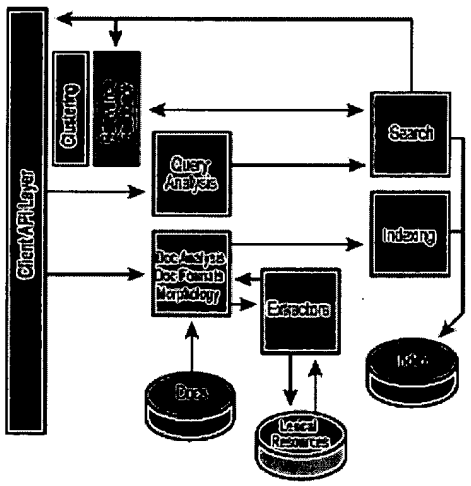


Figure 1. Intelligent Miner for text.

Martin Rennhackkamp is the owner and principal consultant of The Data Base Approach, a corporation specializing in relational and distributed databases, based in Cape Town, South Africa. You can reach Martin via the Internet at mr@dba.co.za.

What did you think of this article? Send a letter to the editor.

Subscribe to *DBMS* -- It's **free** for qualified readers in the United States
August 1998 Table of Contents | Other Contents | Article Index | Search | Site Index | Home

Freeform Search

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term:

Display: 10 **Documents in Display Format:** - **Starting with Number** 1

Generate: ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search Clear Interrupt

Search History

DATE: Tuesday, February 27, 2007 **Purge Queries** Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L24	L22 and (customer with profile or customer near profile or customer adj profile or customer near characteristics or customer with characteristics or customer adj characteristics)	62	L24
L23	L22 not @py>1999	0	L23
L22	L21 and (metadata or meta with data or meta near data or meta adj data or meta-data) and (attributes or subscription and billing and payment and promotion and price with plan and change and service with call and cancellation)	70	L22
L21	l17 and (customer-centric or customer near centric or customer adj centric or customer with centric)	131	L21
L20	L19 and builder	43	L20
L19	l17 and (mapp$ with rules or mapp$ near rules or mapping adj rules)	232	L19
L18	L17 and (virtual near schema or virtual adj schema or virtual with schema)	40	L18
L17	(data with warehouse or "data warehouse" or data with mart or "data mart")	7663	L17
L16	712.clas.	13794	L16
L15	712/1	1025	L15

L14	717/1	1237	L14
L13	717.clas.	13349	L13
L12	705/10	3496	L12
L11	705/1	6844	L11
L10	705.clas.	48536	L10
L9	707.clas.	41392	L9
L8	707/201	3733	L8
L7	707/104.1	7872	L7
L6	707/103y	353	L6
L5	707/102	9424	L5
L4	707/10	14322	L4
L3	707/100	9701	L3
L2	707/3	10188	L2
L1	707/5	4865	L1

END OF SEARCH HISTORY

Terms used **star schema** Found **862** of **198,146**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Industrial sessions: commercial implementation techniques: Efficient execution of joins in a star schema
Andreas Weininger
June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data SIGMOD '02**
Publisher: ACM Press
Full text available: pdf(349.43 KB) Additional Information: full citation, abstract, references, index terms

A star schema is very popular for modeling data warehouses and data marts. Therefore, it is important that a database system which is used for implementing such a data warehouse or data mart is able to efficiently handle operations on such a schema. In this paper we will describe how one of these operations, the join operation --- probably the most important operation --- is implemented in the IBM Informix Extended Parallel Server (XPS).

2 Optimizing large star-schema queries with snowflakes via heuristic-based query rewriting
Yingying Tao, Qiang Zhu, Calisto Zuzarte, Wing Lau
October 2003 **Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research CASCON '03**
Publisher: IBM Press
Full text available: pdf(182.43 KB) Additional Information: full citation, abstract, references, index terms

User queries have been becoming increasingly complex (e.g., involving a large number of joins) as database technology is applied to some application domains such as data warehouses and life sciences. Query optimizers in existing database management systems often suffer from intolerably long optimization time and/or poor optimization results when optimizing large join queries. One possible solution to tackle these problems is to rewrite a user-specified complex query into another form that can be ...

Keywords: complex query, database management system, query graph, query optimization, query rewrite

3 Articles: Reconsidering Multi-Dimensional schemas
Tim Martyn
March 2004 **ACM SIGMOD Record**, Volume 33 Issue 1
Publisher: ACM Press

Full text available: pdf(163.67 KB) Additional Information: full citation, abstract, references, citings

This paper challenges the currently popular "Data Warehouse is a Special Animal" philosophy and advocates that practitioners adopt a more conservative "Data Warehouse=Database" philosophy. The primary focus is the relevancy of Multi-Dimensional logical schemas. After enumerating the advantages of such schemas, a number of caveats to the presumed advantages are identified. The paper concludes with guidelines and commentary on implications for data warehouse design methodologies.

4 Query processing: Implementing operations to navigate semantic star schemas

Alberto Abelló, José Samos, Fèlix Saltor

November 2003 **Proceedings of the 6th ACM international workshop on Data warehousing and OLAP DOLAP '03**

Publisher: ACM Press

Full text available: pdf(193.82 KB) Additional Information: full citation, abstract, references, citings, index terms

In the last years, lots of work have been devoted to multidimensional modeling, star shape schemas and OLAP operations. However, "drill-across" has not captured as much attention as other operations. This operation allows to change the subject of analysis keeping the same analysis space we were using to analyze another subject. It is assumed that this can be done if both subjects share exactly the same analysis dimensions. In this paper, besides the implementation of an algebraic set of operatio ...

Keywords: OLAP operations, SQL, drill-across, semantic relationships, star schema

5 Modelling stars using XML

Jaroslav Pokorny

November 2001 **Proceedings of the 4th ACM international workshop on Data warehousing and OLAP DOLAP '01**

Publisher: ACM Press

Full text available: pdf(2.29 MB) Additional Information: full citation, abstract, references, citings, index terms

We suppose collections of XML data described by Document Type Definitions (DTDs). This data has been generated by applications and plays a role of OLTP database(s). A star schema, a well-known technique used in data warehousing, can be applied. Then dimension information is supposed to be contained in XML data. We will use the notions of subDTD and view, and formulate referential integrity constraints in XML environment. We use simple pattern matching capabilities of current XML query languages ...

Keywords: XML, data warehouse, dimension, star schema

6 Research session: integration and mapping #1: Information preserving XML schema embedding

Philip Bohannon, Wenfei Fan, Michael Flaster, P. P. S. Narayan

August 2005 **Proceedings of the 31st international conference on Very large data bases VLDB '05**

Publisher: VLDB Endowment

Full text available: pdf(241.56 KB) Additional Information: full citation, abstract, references, citings, index terms

A fundamental concern of information integration in an XML context is the ability to *embed* one or more source documents in a target document so that (a) the target document conforms to a target schema and (b) the information in the source document(s) is *preserved.* In this paper, information preservation for XML is formally studied, and the

results of this study guide the definition of a novel notion of *schema embedding* between two XML DTD schemas represented as graphs. Schem ...

7 Star graphics: An object-oriented implementation

Daniel E. Lipkie, Steven R. Evans, John K. Newlin, Robert L. Weissman

July 1982 **ACM SIGGRAPH Computer Graphics , Proceedings of the 9th annual conference on Computer graphics and interactive techniques SIGGRAPH '82**, Volume 16 Issue 3

Publisher: ACM Press

Full text available: pdf(955.07 KB) Additional Information: full citation, abstract, references, citings, index terms

The XEROX Star 8010 Information System features an integrated text and graphics editor. The Star hardware consists of a processor, a large bit-mapped display, a keyboard and a pointing device. Star's basic graphic elements are points, lines, rectangles, triangles, graphics frames, text frames and bar charts. The internal representation is in terms of idealized objects that are displayed or printed at resolutions determined by the output device. This paper describes the design and implementa ...

Keywords: Business graphics, Subclassing

8 Designing data marts for data warehouses

October 2001 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 10 Issue 4

Publisher: ACM Press

Full text available: pdf(203.43 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Data warehouses are databases devoted to analytical processing. They are used to support decision-making activities in most modern business settings, when complex data sets have to be studied and analyzed. The technology for analytical processing assumes that data are presented in the form of simple data marts, consisting of a well-identified collection of facts and data analysis dimensions (star schema). Despite the wide diffusion of data warehouse technology and concepts, we still miss me ...

Keywords: conceptual modeling, data mart, data warehouse, design method, software quality management

9 On relationships offering new drill-across possibilities

Alberto Abelló, José Samos, Fèlix Saltor

November 2002 **Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP DOLAP '02**

Publisher: ACM Press

Full text available: pdf(236.84 KB) Additional Information: full citation, abstract, references, citings, index terms

OLAP tools divide concepts based on whether they are used as analysis dimensions, or are the fact subject of analysis, which gives rise to star shape schemas. Operations are always provided to navigate inside such star schemas. However, the navigation among different stars is usually overlooked. This paper studies different kinds of Object-Oriented conceptual relationships (part of UML standard) between stars (namely *Derivation*, *Generalization*, *Association*, and *Flow*) ...

Keywords: UML, drill-across, multidimensional design, semantics

10 Heuristic optimization of OLAP queries in multidimensionally hierarchically clustered databases

Dimitri Theodoratos, Aris Tsois

November 2001 **Proceedings of the 4th ACM international workshop on Data warehousing and OLAP DOLAP '01**

Publisher: ACM Press

Full text available: pdf(1.44 MB) Additional Information: full citation, abstract, citings, index terms

On-line analytical processing (OLAP) is a technology that encompasses applications requiring a multidimensional and hierarchical view of data. OLAP applications often require fast response time to complex grouping/aggregation queries on enormous quantities of data. Commercial relational database management systems use mainly multiple one-dimensional indexes to process OLAP queries that restrict multiple dimensions. However, in many cases, multidimensional access methods outperform one-dimensiona ...

11 Schemata for interrogating solid boundaries

Michael Karasick, Derek Lieber

May 1991 **Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications SMA '91**

Publisher: ACM Press

Full text available: pdf(937.48 KB) Additional Information: full citation, references, index terms

12 Database component ware

Bernhard Thalheim

January 2003 **Proceedings of the 14th Australasian database conference - Volume 17 ADC '03**

Publisher: Australian Computer Society, Inc.

Full text available: pdf(301.28 KB) Additional Information: full citation, abstract, references, citings, index terms

Database modeling is still a job of an artisan. Due to this approach database schemata evolve by growth without any evolution plan. Finally, they cannot be examined, surveyed, consistently extended or analyzed. Querying and maintenance become very difficult. Distribution of database fragments becomes a performance bottleneck. Currently, databases evolve to huge databases. Their development must be performed with the highest care.This paper aims in developing an approach to *systematic* sch ...

13 Event-entity-relationship modeling in data warehouse environments

Lars Bækgaard

November 1999 **Proceedings of the 2nd ACM international workshop on Data warehousing and OLAP DOLAP '99**

Publisher: ACM Press

Full text available: pdf(634.98 KB) Additional Information: full citation, abstract, references, citings, index terms

We use the event-entity-relationship model (EVER) to illustrate the use of entity-based modeling languages for conceptual schema design in data warehouse environments. EVER is a general-purpose information modeling language that supports the specification of both general schema structures and multi-dimensional schemes that are customized to serve specific information needs. EVER is based on an event concept that is very well suited for multi-dimensional modeling because measurement data oft ...

Keywords: data warehousing, event modeling, information modeling, multi-dimensional modeling, star schemes

14 XPS a database server for data warehousing

Andreas Weininger
November 2001 **Proceedings of the 4th ACM international workshop on Data warehousing and OLAP DOLAP '01**
Publisher: ACM Press
Full text available: pdf(621.63 KB) Additional Information: full citation, abstract, citings, index terms

A database server used for implementing a data warehouse must support other features than a database server used for OLTP. Therefore, in this paper we will look specifically at features necessary for efficiently processing queries on a database with a star schema model, a database scheme which is used very often in data warehousing. We will especially analyze the features provided for this by the IBM Extended Parallel Server (XPS). There are special star join methods like the Push-Down Hash Semi ...

15 Dynamic maintenance of multidimensional range data partitioning for parallel data processing

Junping Sun, William I. Grosky
November 1998 **Proceedings of the 1st ACM international workshop on Data warehousing and OLAP DOLAP '98**
Publisher: ACM Press
Full text available: pdf(1.09 MB) Additional Information: full citation, references, citings, index terms

16 Query processing: Exploiting hierarchical clustering in evaluating multidimensional aggregation queries

Dimitri Theodoratos
November 2003 **Proceedings of the 6th ACM international workshop on Data warehousing and OLAP DOLAP '03**
Publisher: ACM Press
Full text available: pdf(216.79 KB) Additional Information: full citation, abstract, references, index terms

Multidimensional aggregation queries constitute the single most important class of queries for data warehousing applications and decision support systems. The bottleneck in the evaluation of these queries is the join of the usually huge fact table with the restricted dimension tables (*star-join*). Recently, a multidimensional hierarchical clustering schema for star schemas is suggested. Subsequently, query evaluation plans for multidimensional queries appeared that essentially implement a ...

Keywords: multidimensional aggregation query, multidimensional hierarchical clustering, query transformations, star join

17 Component-driven engineering of database applications

Klaus-Dieter Schewe, Bernhard Thalheim
January 2006 **Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling - Volume 53 APCCM '06**
Publisher: Australian Computer Society, Inc.
Full text available: pdf(188.64 KB) Additional Information: full citation, abstract, references, index terms

Though it is commonly agreed that the design of large database schemata requires group effort, database design from component subschemata has not been investigated thoroughly. In this paper we investigate snowflake-like subschemata of database schemata expressed in the Higher-order Entity-Relationship Model (HERM). These subschemata are almost hierarchical in the sense that they may contain cycles in the schema, but not in the instances. We show that each HERM schema can be decomposed

into such ...

18 A comparison of data warehousing methodologies

Arun Sen, Atish P. Sinha
March 2005 **Communications of the ACM**, Volume 48 Issue 3
Publisher: ACM Press
Full text available: pdf(117.81 KB) html(28.41 KB) Additional Information: full citation, abstract, references, index terms

Using a common set of attributes to determine which methodology to use in a particular data warehousing project.

19 Requirement-based data cube schema design

David W. Cheung, Bo Zhou, Ben Kao, Hongjun Lu, Tak Wah Lam, Hing Fung Ting
November 1999 **Proceedings of the eighth international conference on Information and knowledge management CIKM '99**
Publisher: ACM Press
Full text available: pdf(1.02 MB) Additional Information: full citation, abstract, references, citings, index terms

On-line analytical processing (OLAP) requires efficient processing of complex decision support queries over very large databases. It is well accepted that pre-computed data cubes can help reduce the response time of such queries dramatically. A very important design issue of an efficient OLAP system is therefore the choice of the right data cubes to materialize. We call this problem the data cube schema design problem. In this paper we show that the problem of finding an op ...

Keywords: DSS, OLAP, data cube schema design, data cubes

20 Industrial, applications, and experience sessions: Industry 2: Decision support: The making of TPC-DS

Raghunath Othayoth, Meikel Poess
September 2006 **Proceedings of the 32nd international conference on Very large data bases - Volume 32 VLDB'2006**
Publisher: VLDB Endowment
Full text available: pdf(658.32 KB) Additional Information: full citation, abstract, references, index terms

For the last decade, the research community and the industry have used TPC-D and its successor TPC-H to evaluate performance of decision support technology. Recognizing a paradigm shift in the industry the Transaction Processing Performance Council has developed a new Decision Support benchmark, TPC-DS, expected to be released this year. From an ease of benchmarking perspective it is similar to past benchmarks. However, it adjusts for new technology and new approaches the industry has embarked o ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Sign in

Google

Web Images Video News Maps more »

data warehouse "star schema" >1998 Search Advanced Search Preferences

Web Results **1** - **10** of about **37,800** for **data warehouse "star schema" >1998**. (**0.13** seconds)

Scholarly articles for **data warehouse "star schema" >1998**

Building a data warehouse for decision support - Poe - Cited by 117
An overview of data warehousing and OLAP technology - Chaudhuri - Cited by 960
starER: a conceptual model for data warehouse design - Tryfona - Cited by 86

Why is the **Star Schema** a Good **Data Warehouse** Design? (ResearchIndex)
@misc{ levene-why, author = "Mark Levene and George Loizou", title = "Why is the **Star Schema** a Good **Data Warehouse** Design? ...
citeseer.ist.psu.edu/levene99why.html - 21k - Cached - Similar pages

Star/Snow-flake Schema Driven Object-Relational **Data Warehouse** ...
The conventional **star schema** model of **Data Warehouse** DW has its limitations due to the nature of ... 3: Modeling Large Scale OLAP Scenarios - Lehner - **1998** ...
citeseer.ist.psu.edu/415858.html - 24k - Cached - Similar pages

[Paper] Multidimensional Partitioning of a **Data Warehouse Star Schema**
Figure 1 - Typical **data warehouse star schema**. The **star schema** has one fact ... One Dimensional Round-Robin Partitioning Model Year P1 P2 P3 P4 **1998** Jan Feb ...
www.actapress.com/PDFViewer.aspx?paperId=25124 - Similar pages

Kimball Group: **Data Warehouse** Training: Books
John Wiley & Sons, **1998** (771 pages) This book by Ralph Kimball, Laura Reeves, ... Mastering **Data Warehouse** Aggregates: Solutions for **Star Schema** Performance ...
www.kimballgroup.com/html/books.html - 22k - Cached - Similar pages

DBMS - August **1998** - Reaping the Web for Your **Data Warehouse**
As we move toward loading **data** into a **warehouse**, the structures must be compatible with the **star-schema** design and with key identifier values. ...
www.dbmsmag.com/9808d14.html - 22k - Cached - Similar pages

DBMS - August **1998** - Server Side
You can populate the **star schema** with **data** processed by the Essbase ... Visual **Warehouse** is IBM's **data warehouse** and **data** mart solution targeted at the ...
www.dbmsmag.com/9808d17.html - 22k - Cached - Similar pages

[PDF] **Data Warehousing** and Decision Support at the National Agricultural ...
File Format: PDF/Adobe Acrobat
star schema because of the nature of the tables and their intended use. ... Adamson, C., & Venerable, M. (**1998**). **Data warehouse** design solutions. ...
ssc.sagepub.com/cgi/reprint/18/4/434.pdf - Similar pages

Data Warehousing
We decided that we could afford this because the conventional wisdom in the **data warehousing** business in **1998** was that up to billion-row fact tables were ...
philip.greenspun.com/sql/**data**-warehousing.html - 81k - Cached - Similar pages

Amazon.com: Oracle8 **Data Warehousing** (Oracle Press): Books ...

They discuss what **data warehousing** is, who needs it, building teams, ... chapters outlining usage of the **star schema**), and implementation of the **data** mart. ...

www.amazon.com/Oracle8-**Data**-Warehousing-Oracle-Press/dp/0078825113 - 102k -
Cached - Similar pages

Papers and Briefs on **Data Warehousing**, **Data** Mining, and OLAP
Six, November 12, **1998**). **Data Warehousing** has lately been undergoing substantial ...
Architectural Evolution in **Data Warehousing** (September 9, **1998**). ...
www.dkms.com/dwdmolappapers.htm - 21k - Cached - Similar pages

Result Page: **1** 2 3 4 5 6 7 8 9 10 **Next**

Executive Information Systems, Inc.

Home of The New Knowledge Management, The Open Enterprise, and

Don't forget EIS Books and Professional Papers!

Papers, Briefs, and Presentations On Data Warehousing, Data Mining, and OLAP

The papers, briefs, and presentations focus primarily on the above subjects and secondarily on Knowledge Management, portal, or DKMS considerations. In instances w this distinction is not so clear, an abstract may appear both here and on the KM Papers p The following white papers, briefs, and presentations on Data Warehousing, Data Mining, OLAP are available.

DW Papers and DKMS Briefs and Presentations on Data Warehousing and OLAP

A Systems Approach to Dimensional Modeling in Data Marts (White Paper No. March 1997). This paper views dimensional modeling in data marts from the viewpoint o Fast Analysis of Shared Multidimensional Information (FASMI) definition of OLAP. F/ implies that dimensional data modeling supports measurement, causal, and struc modeling, as well as a specification and organization of data comprehensive enougl supporting KDD. The paper attempts to support this need by offering an approac dimensional modeling designed to support the development of business area mode system dynamics. The approach requires highly explicit, top-down conceptualization data inventory steps in order to sketch out the broadest possible view of the outlines o range of measurement and cause-effect relations underlying data marts.

Data Mining and KDD: A Shifting Mosaic (White Paper No. Two, March 1997). Mining as a field is not yet through with the process of definition and conceptualization o scope of the field. There are at least three distinct concepts of data mining being use practitioners and vendors. This paper defines the three concepts, associates them with t related concepts of Knowledge Discovery in Databases (KDD), and argues that data mini not automatic knowledge discovery, and that the dream of making it so is, at best, an motivating long-term development.

Data Warehouses and Data Marts: A Dynamic View (White Paper No. Three, M 1997). This paper explores three patterns of data mart development and relationships data warehouses: the top-down model; the bottom-up model; and the parallel develop model. All three models are seen as unrealistic because they view development wit explicit consideration of user feedback and its impact on development. Three related m

in the presence of user feedback are then presented, their dynamics are discussed, some predictions are made about the likely popularity of each of the three feedback m in the future.

Evaluating OLAP Alternatives (White Paper No. Four, March 1997). The rush to de data warehouses and data marts has gained considerable momentum from the presen server-based On-line Analytical Processing (OLAP) tools, including: Multidimensional Se based (MDOLAP) tools; a number of Relational OLAP (or ROLAP) products; and a new called Sybase IQ which uses a technology we can call Vertical Technology OLAP (VTOI How do we choose an OLAP product for a data warehouse or data mart? This White Pape reviews the three OLAP product categories, and (b) provides a set of criteria for pr evaluation in specific product contexts.

Object-Oriented Data Warehousing (White Paper No. Five, August 1997). warehousing has largely developed with little or no reference toObject-Oriented Soft Engineering (OOSE). This is consistent with its development out of two-tier client/s relational database methodology. As data warehouses increasingly are supplemented data marts, with data stores of diverse type and content, and with internet and int front-ends, the two-tier client/server paradigm has given way to a multi-tier conceptua software framework characterized by distributed objects. Multi-tiered data wareho needs to be reconceptualized in terms of distributed objects and therefore in terms of O This paper offers such a reconceptualization with a focus on dimensional data modelin its relation to object modeling.

Dimensional Object Modeling (White Paper No. Seven, April 30, 1998). An o modeling approach offers advantages in supporting Dimensional Data Modeling (DD data warehouses and data marts. The current approach to making the basic decisio producing a DDM is a pragmatic one. The pragmatic approach has had conside commercial success, but it still makes tight coupling of strategic goals and objectives t DDM result a matter of art, rather than a product of an explicit method or procedure, re in a model composed of passive containers for data attributes, rather than components combine both data and behavior,does not place DDM within a broader framewor integrating data and process -- that is, the pragmatic approach is too data-centric, at a when data warehousing is concerned with integrating a complex diversity of server-b decision support system functions. This paper examines the nature of DDM and develops the argument for tight coupling of strategic goals and objectives to the through an object modeling approach, and discusses the advantages of the DOM approa more detail.

Dimensional Modeling and E-R Modeling in the Data Warehouse (White Pape Eight, June 22, 1998). While there is consensus in the field of data warehousing or desirability of using DM/star schemas in developing data marts, there is an on-g controversy over the form of the data model to be used in the data warehouse. "Inmonites" contend that the data warehouse should be developed using an E-R model. "Kimballites" believe that the data warehouse should always be modeled using a DM schema. Indeed Kimball has stated that while DM/star schemas have the advantag greater understandability and superior performance relative to E-R models, their use inv no loss of information, because any E-R model can be represented as a set of DM schema models. This paper discusses two issues related to the controversy. First, the that any E-R model can be represented as an equivalent set of DM/star schema models, second, the question of whether an E-R structured data warehouse, absent associ entities, i.e. fact tables, is a viable concept, given recent developments in data warehous

Architectural Evolution in Data Warehousing (White Paper No. Eleven, July 1, 1

This paper is concerned with DSS/data warehouse system architectural evolution in resp to the growing complexity of the enterprise DSS environment and with the relationsh new architectures to a developing capability to handle the Dynamic Integration Problem paper briefly describes and analyzes the following architectures: Top-Down; Bottom Enterprise Data Mart (EDM); Data Stage/Data Mart (DS/DM); Distributed Warehouse/Data Mart (DDW/DM); Distributed Knowledge Management (DKM); Varia with introduction of the ODS. In addition it comments on the relationship between architecture and data mining and provides some brief comments on software tool implementing DKMA.

DKMS Briefs

The Corporate Information Factory or the Corporate Knowledge Factory? (D Brief No. One, July 10, 1998). Bill Inmon has introduced the Corporate Information Fac But should he have introduced the Corporate Knowledge Factory? Does it really make difference?

Is Data Staging Relational? A Comment (DKMS Brief No. Five, November 11, 199 recent question raised by Ralph Kimball is whether the data staging area is relational o more to do with sequential processing of flat files. This brief revisits and expand Kimball's viewpoint. It examines the above issue from the viewpoint of the data sta application server, the data staging repository, and the metadata and metamodel that the data staging process.

Data Warehouses, Data Marts, and Data Warehousing: New Definitions and Conceptions (DKMS Brief No. Six, November 12, 1998). Data Warehousing has lately undergoing substantial changes in architecture and a broadening of related funct applications. With these changes have come new definitions of the Data Warehouse evolving conceptions of data warehousing This brief explores a number of data wareh and data mart definitions and their relation to the idea of the Distributed Knowl Management System (DKMS). It also analyzes the meaning of "data warehousing," in lig changes in data warehousing systems and changes in definitions.

DKMA and The Data Warehouse Bus Architecture (DKMS Brief No. Seven, Nove 13, 1998). The Data Warehouse Bus Architecture is composed of "a master suit conformed dimensions" and standardized definitions of facts. Business process data r throughout an enterprise can "plug into" this bus to receive the dimension and fact t they need. The Bus supports the various processes and associated data marts that mea key aspects of the processes. The logical union of these data marts is said to be the warehouse. And each data mart is said to be a subset of that data warehouse. This p describes the Data Warehouse Bus Architecture offered by Kimball, Reeves, Ross, Thornthwaite, and then contrasts it with DKM Architecture, an object-oriented alternative

Presentations

Architectural Evolution in Data Warehousing (September 9, 1998).

The application of Web technology to data warehousing is an intriguing combination, but does it deliver practical value?

by Richard Hackathorn

Reaping the Web for Your Data Warehousi

DBMS, August 1998

Amid the chaos of the Web is a diverse collection of ever-changing information, some of which can be highly valuable to your enterprise. The challenge is to wade (with big boots) through the Web, discovering and acquiring those resources that have value for your business. In this article, I will explore the use of information resources from the Web as input to data warehouses. I call this activity Web farming, or the systematic refining of information resources on the Web for business intelligence.

Both the Web and data warehousing are hot technologies receiving considerable attention within the IT industry. In several areas, the combination has proven highly successful. Publishing warehouse data via an intranet is a highly productive approach that combines Web delivery mechanisms with Web-enabled databases. Generating dynamic pages from Web-enabled databases and adding Java applets to manipulate data locally to the browser has made available whole new areas of data analysis and data mining for warehouse users.

In contrast, no one has seriously considered extracting content from the Web and using it as input to the data warehouse. The paradigm of the Web is radically different from the paradigm for the data warehouse. Adapting an old programming term, you might say that Web content is spaghetti data. That is, it links to everywhere with little discipline. Furthermore, Web content is highly volatile and constantly changing. The Web's diversity challenges our imagination and appreciation for new forms of creative expression. The problem is cultivating those few nuggets with real business value from that diversity.

Reactions to using Web content tend to be negative. Web content is too unreliable and unstable for business decisions. The interaction with Web sites is too messy. Transformation of hypertext into a structured database is often impossible. Images and sound contain a lot of hidden content but are not discernible to a machine.

There are increasing instances of various systems integrating Web content into their operations. A simple example is the monitoring of international currency rates by a U.K. financial firm. The system monitors three specific financial sites for changes, retrieves the contents, and parses the retrieved page to obtain the useful data. With a delay of approximately 20 minutes, it retrieves currency exchange rates (such as USD to GBP) along with stock prices and news headlines containing predefined keywords. The data is loaded into several database tables, along with linking information, source, and a date/time

stamp. The user can choose the currency for display and ask for recent headlines concerning that currency.

There are hundreds of commercial databases available via the Web, some requiring substantial fees. An example of a recent addition is the IBM Web site (patent.womplex.ibm.com/respage.html) that offers a database of patents issued in the United States during the last 27 years - more than 2 million patents. You can visually display the results, highlighting clusters of similar patents or ranking patents according to areas of interest. The imperative exists to incorporate the analysis of patent information in firms of all sizes, including R&D, competitive intelligence, licensing management, and strategic planning. Even the investment banking community can determine the intellectual assets of potential acquisition or investment candidates.

New approaches to handling Web content systematically are emerging weekly. The Junglee Corp. (www.junglee.com) has commercialized research at Stanford University to turn the Web into a single virtual database. Using specific wrappers around Web sources, unified metadata drives a database engine to process queries joining multiple sources. This technology has been applied to job classified ads (for example, JobCanopy in the San Jose area) and e-commerce (ShopCanopy for 40 merchants in eight categories).

Using the Web for Business Intelligence

Many people think that data external to the organization has little value to the business because internal operational systems contain all the required data. Is there a need for companies to integrate external data into their warehouses?

Professor Peter Drucker, a senior guru of management practice, admonishes IT executives to look outside their enterprises for information. He remarked that the single biggest challenge is to "organize outside data because change occurs from the outside." He predicted that the obsession with internal data would lead to organizations being blindsided by external forces.

In the majority of data warehousing efforts, enterprises focus inward. As markets become turbulent, the traditional way of doing business becomes less viable. Data from internal operational systems becomes less relevant to managing your business and planning for its future. Instead, the enterprise should be keenly alert to outside sources.

An enterprise must know more and more about its customers, suppliers, competitors, government agencies, and other external factors. It must enhance the information from internal systems with information about external factors. The synergism of the combination creates the greatest business benefit for the enterprise.

The Web has become the universal and global delivery mechanism for external data. In many ways, the Web is the mother of all data warehouses. The immense resources of the Web, with all of its complexity and dynamics, are largely untapped. Valuable information about external business factors is readily available on the Web and is becoming more so each day.

Objectives of Web Farming

Web farming is not surfing the Web haphazardly, wandering from one intriguing item to another. Nor is it a one-time search of the Web. On a continuous and systematic basis, a Web farming system must deliver, to the right people at the right time, information highly relevant to the enterprise. In effect, a

Web farming system acts as the eyes and ears of the enterprise, focusing externally to be aware of important changes in the business environment.

Web farming has the objective of refining Web content in a systematic manner. In particular, refining this content involves the processes of discovering, acquiring, structuring, and disseminating, as I'll explain later. Therefore, the specific objectives of Web farming are:

- To discover Web content that is highly relevant to the business
- To acquire that content so it is properly validated within a historical context
- To structure the content into a useful form that's compatible with the data warehouse
- To disseminate the content to the proper people so it has direct and positive impacts on specific business processes
- To manage the previous steps in a systematic manner as part of the production operations of a data center environment.

Web farming is often confused with our personal experiences of surfing the Web -- long periods of frustration with a few moments of elation. However, Web farming is serious business. Many people falsely think that Web farming is like planting a small garden in the backyard. In contrast, Web farming is like managing a large agricultural concern that involves many people and several thousand acres of farmland. There are similarities in the basic concepts, but the scaling of a personal garden to an agricultural business changes the methodology, architecture, tools, and techniques.

Reliability of Web Content

The reliability of Web content is an important issue that you must manage carefully. Consider the following situation. If you hear, "Buy IBM stock because it will double over the next month," your reaction should depend on who made that statement and in what context. Was it a random conversation overheard on the subway, a chat with a friend over dinner or a phone call from a trusted financial advisor? The same is true with judging the reliability of Web content.

Most people have the "flake free" image of Web content. In reality, the Web is a global bulletin board where the wise and the foolish have equal space. Acquiring content from the Web should not reflect positively or negatively on its quality.

Think of Web resources in terms of quality and coverage. (See Figure 1) Toward the top are information resources of high quality (for example, accuracy, currency, and validity), while resources toward the right have a wide coverage (for example, scope, variety, and diversity). The interesting aspect of the Web is that its information resources occupy all the quadrants in this figure.

In the upper center of the figure, the commercial online databases from Dialog Information Services and similar vendors have traditionally supplied businesses with high-quality information about numerous topics. However, the complexity of using these services and the infrequent update cycles have limited their usefulness.

To the left, government databases have become tremendously useful in recent years. Public information was often available only by spending many hours of manual labor at libraries or government offices. The Electronic Data Gathering, Analysis, and Retrieval (EDGAR) database maintained by the U.S. Security and Exchange Commission contains extensive information on publicly traded companies and is updated daily.

In the upper left, corporate Web sites often contain vast amounts of useful information in white papers, product demos, and press releases, eliminating the necessity to attend trade exhibits to learn the "latest and greatest" in a marketplace.

Finally, the flaky content occupies the lower half of the figure. Its value is not in the quality of any specific item but in its constantly changing diversity. In combination with the other Web resources, the flaky content acts as a wide-angle lens to avoid tunnel vision of one's marketplace.

Information Flow

The data warehouse occupies a central position in the information flow of a Web farming system. Like operational systems, the Web farming system provides input to the data warehouse. The result is to disseminate the refined information about specific business subjects to the enterprise.

As the primary source of external perspectives on the business, the Web can be supplemented (but not replaced) by content from the enterprise's intranet. This content is typically in the format of internal Web sites, word processing documents, spreadsheets, and email messages. However, the content from the intranet is usually limited to internal information about the enterprise, missing an important aspect of Web farming.

Most information acquired by the Web farming system will not be in a form suitable for the data warehouse. It will either be unstructured hypertext or unverified tabular values. In either case, you must perform a process of refining that information before loading it into the warehouse. Even in this unrefined state, this information could be highly valuable to the enterprise. It may be useful to disseminate this information directly via textual message alerts or "What's New" bulletins.

Refining Information

When a data warehouse is first implemented within an enterprise, you need to analyze and reengineer the data from operational systems. The same is true for Web farming. Before you can load Web content into a warehouse, you must refine that information.

There are four processes for refining information: discovery, acquisition, structuring, and dissemination.

Discovery is the exploration of available Web resources to find those items that relate to specific topics. Discovery involves considerable "detective" work far beyond searching generic directory services (such as Yahoo) or indexing services (such as AltaVista). Furthermore, the discovery activity must be a continuous process because data sources are continually appearing (and disappearing) from the Web. A business analyst is the central figure in this activity and requires advanced search and indexing tools to be productive.

Acquisition is the collection and maintenance of content identified by its source. The main goal of acquisition is to maintain the historical context so you can analyze content in the context of past changes. Acquisition requires a secured server platform with large storage capacity.

Structuring is the analysis, validation, and transformation of content into a more useful format and into a more meaningful structure. The formats can be Web pages, spreadsheets, word processing documents, and database tables. As we move toward loading data into a warehouse, the structures must be compatible with the star-schema design and with key identifier values.

Dissemination is the packaging and delivery of information to the appropriate consumers, either directly or through a data warehouse. It requires a range of dissemination mechanisms from predetermined schedules to ad hoc queries. Newer technologies such as information brokering and preference matching may be desirable.

There is a bidirectional flow to the processes. (See Figure 2.) The left-to-right flow refines the content of information, which becomes more structured and validated. The right-to-left flow refines the control of the processes, which become more selective and discriminating.

Rendezvous with the Data Warehouse

The most difficult part of Web farming is the rendezvous with the data warehousing system, especially in matching the data structure of Web content with the data warehouse schema.

Consider a simple data schema for a sales warehouse. (See Figure 3a.) In this warehouse, we have sales data by customer, product, and store aggregated on a weekly basis. Let's assume that we have mostly corporate customers, rather than individuals, as in a large office furniture company.

Web farming would be valuable by enhancing the demographics (for example, quarterly financials) about customers, such as you can find in the EDGAR Web site. (See Figure 3b.) By adding information on customer demographics, you can perform selective marketing based on the profitability and requirements of customers. By knowing what types of customers buy what types of products at which stores, we can promote specific sales and anticipate demand. For example, companies that are expanding are more likely to order office furniture.

Demographic information is added to the customer dimension to enhance analyses. As experience with the demographics matures, data mining techniques can cluster customers into meaningful categories based on demographics. (See Figure 3c.)

Another example of using Web farming to enhance a data warehouse is the addition of demographics on the store. (See Figure 3d.) Using ZIP codes and even the full street address, you can add census data about the communities surrounding the store to your data warehouse as another business dimension. This enhancement can lead to more effective management of stores based on their communities and more effective placement of new stores.

A final example involves adding data that is highly volatile, such as weather. (See Figure 3e.) Seasonal variations have always been an important part of sales analysis. However, a sudden heavy snowstorm or an intense hailstorm can also affect sales of specific products in addition to the seasonal variations. This example shows that timely and continuous flow of Web content into the warehouse can aid in the day-to-day management of the business.

Where Are We Heading?

In many ways, the data warehouse is not a requirement for Web farming. You could successfully farm the Web, reaping tremendous value for the business and bypass the data warehouse entirely. However, establishing the Web farming function is much easier for an enterprise if it has a mature understanding of data warehousing and several successful experiences with data warehousing.

Across the industry, the current practice of data warehousing is fulfilling its promises of business benefits. In retrospect, the current benefits from data warehousing are "low-lying fruit" -- easy

accomplishments (relatively speaking) of purging the sins of monolithic legacy systems. Web farming will challenge us with deeper issues concerning information refinement and knowledge management.

Web farming will be an agent of change (even of a disruptive sort) to the controlled and structured world of data warehousing. This is a necessary change -- a maturing of the basic objectives of data warehousing into a more comprehensive system of knowledge management for the enterprise.

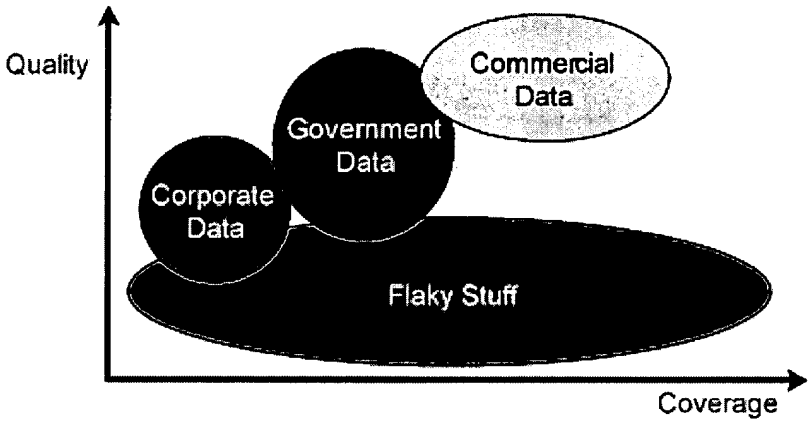


Figure 1. Web resources in terms of quality and coverage.

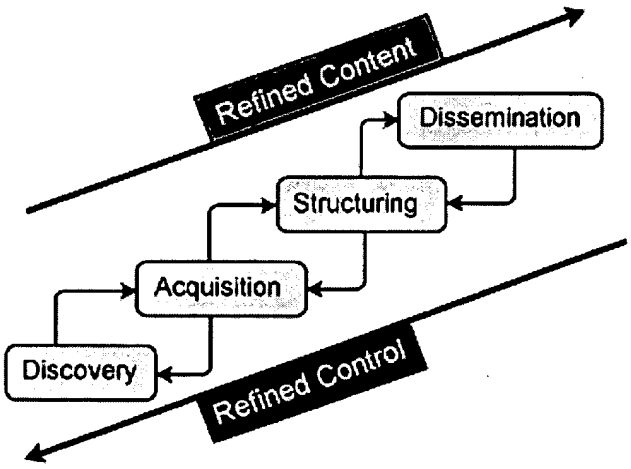


Figure 2. Refining information.

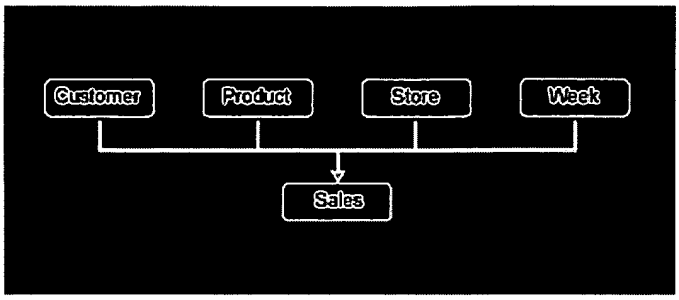


Figure 3a. Typical data schema for a sales warehouse.

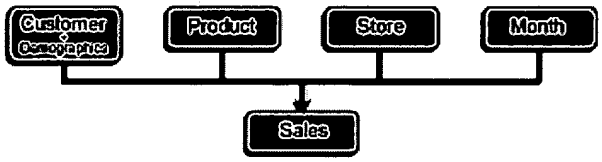


Figure 3b. Adding customer demographics - Part I.

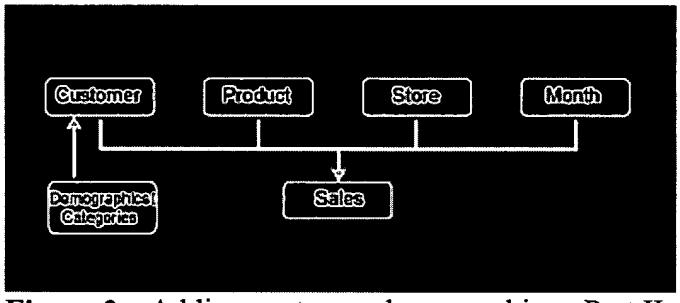


Figure 3c. Adding customer demographics - Part II.

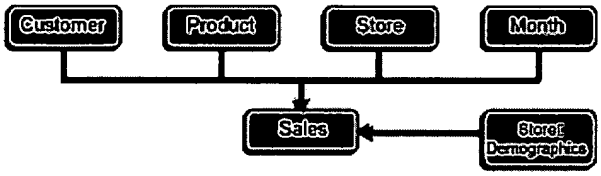


Figure 3d. Adding store demographics.

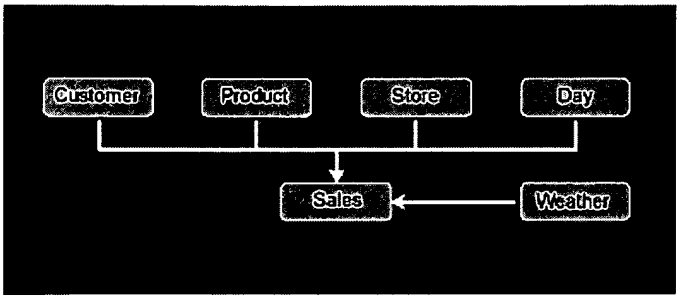


Figure 3e. Adding weather data.

Richard Hackathorn is president and founder of Bolder Technology Inc., a firm specializing in enterprise connectivity and data warehousing in Boulder, Colo. You can reach him at richardh@bolder.com. His Web farming resource site is located at webfarming.com.

Note: This article is based upon excerpts from the forthcoming book entitled Web Farming for the Data Warehouse *to be published by Morgan Kaufmann Publishers this fall.*

What did you think of this article? Send a letter to the editor.

Subscribe to *DBMS* -- It's **free** for qualified readers in the United States
August 1998 Table of Contents | Other Contents | Article Index | Search | Site Index | Home

---

Please send questions or comments to dbms@mfi.com
Updated July 7, 1998

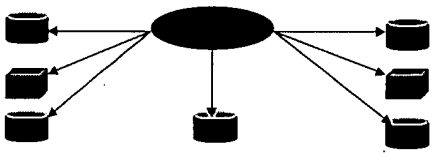
Executive Information Systems, Inc.

Architectural Evolution in Data Warehousing: The Coming of Distributed Knowledge Management Architecture (DKMA)

By

Joseph M. Firestone, Ph.D.
eisai@moon.jic.com

September 9, 1998



EIS Web Site: http://www.dkms.com

Dynamic Integration and Data Warehousing

- The Dynamic Integration Problem is the problem of proactively and automatically monitoring and managing evolutionary change in data warehousing systems without imposing a traditional and constraining "Top-Down" architecture.
- It is the problem of providing managers of both data warehouses and data marts the power to innovate, while still maintaining the integration and consistency of the system.

Dynamic Integration and Architectural Evolution in Data Warehousing

- Data Warehousing is now a complex systems integration problem. A full-blown Data Warehousing System may encompass
  - the following database servers:
    - The data warehouse
    - various data marts (department, function, or application-specific DSSs, using Relational, Multi-dimensional (MOLAP), or Column-based Servers)
    - One or more Operational Data Stores (ODSs)
    - One or more Data Staging Areas

Dynamic Integration and Architectural Evolution in Data Warehousing (Two)

- the following application servers
  - Web Servers
  - ETML Servers
  - Data Mining servers
  - Stateless Transaction Servers (e.g., MTS, Jaguar CTS, etc.
  - Business Process engines (e.g. Persistence Power-Tier, DAMAN InfoManager)
  - Document Servers
  - ROLAP Support Servers (e.g., MicroStrategy, Information Advantage)
  - Report Servers
- and various front-end OLAP and reporting tools

Dynamic Integration and Architectural Evolution in Data Warehousing (Three)

- The Dynamic Integration problem in the context of this complexity of components and interactions is three-fold:
  - First, we need an integrated view of all server-based assets;
  - Second, we need to manage flows of data, information, and knowledge throughout this system to maintain the common view in the face of change in form and content, and to distribute the system's data, information, and knowledge bases as required, and

Dynamic Integration and Architectural Evolution in Data Warehousing (Four)

- Third, we need such management to occur automatically and without centralizing the system so that the authority and responsibility for adding new data and information to the system is distributed.
- Automated dynamic integration is a capability not now provided by data warehousing vendors. It is increasing recognition of the need for this capability that drives architectural evolution in the DW system.

Architectures for Managing DSS Integration

- Here are six architectures for managing and viewing the problem of integration.
- Top-Down Architecture
  (Inmon, Prism Solutions, Carleton Corporation, ETI)
- Bottom-Up Architecture
  (Broadbase Information Systems, Sagent, Ardent DataStage)
- Enterprise Data Mart Architecture (EDMA)
  (Informatica, Carleton)
- Data Stage/Data Mart Architecture (DS/DMA)
  (Informatica, Carleton, Sagent)

Architectures for Managing DSS Integration (Two)

- Distributed Data Warehouse/Data Mart Architecture (DDW/DMA) *(Sybase, Platinum (HP) Intelligent Warehouse)*
- Distributed Knowledge Management Architecture (DKMA) *(The sole vendor targeting this architecture is DAMAN Consulting)*
- *There are variations of each architecture incorporating an ODS*

Top-Down Architecture

- The first Data Warehousing Systems architecture
- Begins with Extraction, Transformation, Migration, and Loading (ETML) process
- Establishes the data warehouse first, along with centralized metadata repository
- Data Marts are constituted from extracted and summarized data warehouse data and metadata
- The Data Warehouse has atomic layer and detailed historical data

Top-Down Architecture (Two)

- The Data Warehouse uses Normalized E-R Models
- Data Marts have highly and lightly summarized data
- Integration is automatic as long as the discipline of constituting data marts as subsets of the data warehouse is maintained
- Tools exist to generate data marts from the data warehouse "by pushing a button"

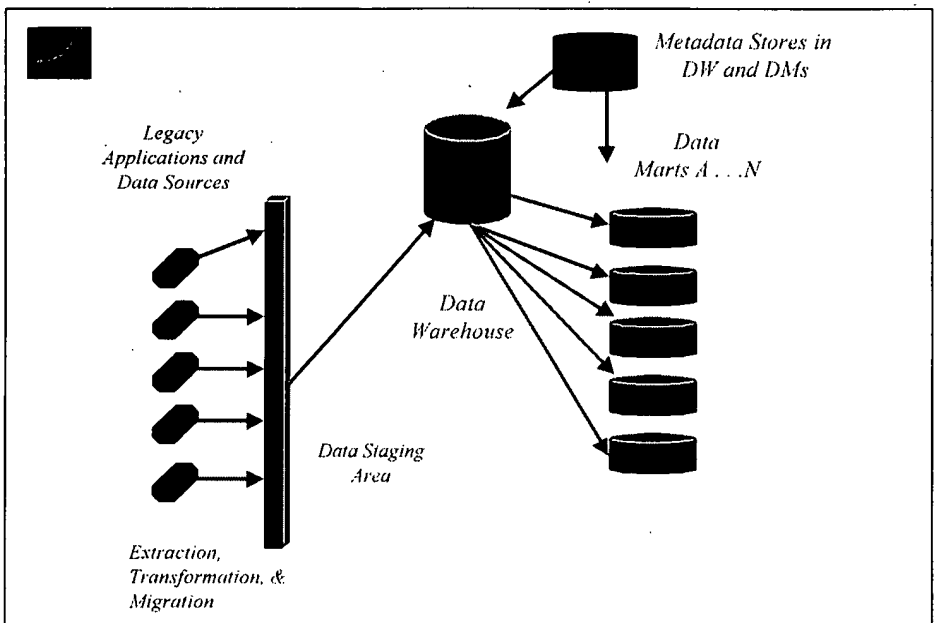


Figure One -- Top-Down Architecture

Bottom-Up Architecture

- The second Data Warehousing Systems architecture
- Became popular because the Top-down architecture took too long to implement, was often politically unacceptable, and was too expensive
- Begins with ETML for one or more Data Marts
- Requires no common data staging area
- Uses Dimensional Data models for Data Marts
- Uses Atomic, lightly summarized, and highly summarized data

Bottom-Up Architecture (Two)

- Provides no common metadata
- Constructs thedata warehouse incrementally over time from data marts
- Adopted initially by second generation tool vendors Informatica, Sagent, and Ardent
- Met expectations in building data marts, but soon was perceived as unacceptable for the long term because lacking common metadata, it is difficult to construct the data warehouse, and it also leads to new "stovepipes" or"legamarts" over time.

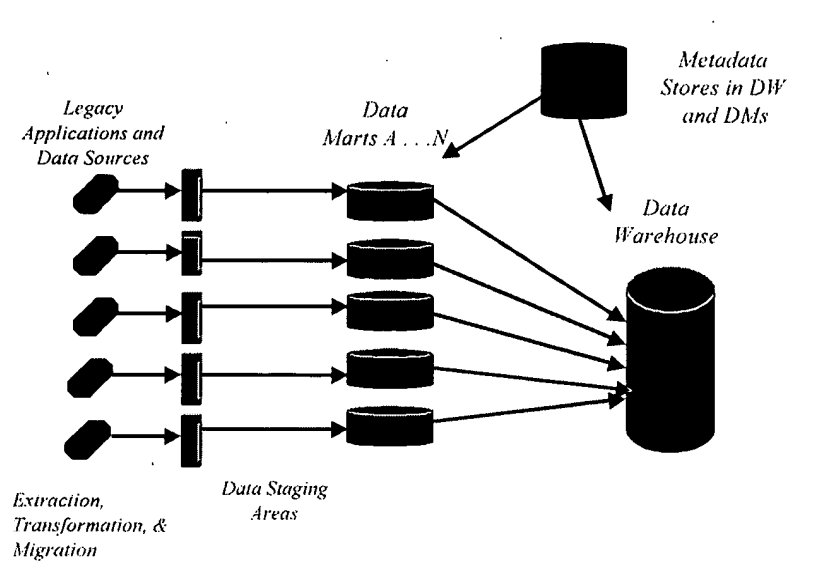


Figure Two -- Bottom-Up Architecture

Enterprise Data Mart Architecture (EDMA)

- A response of the Bottom-up supporters to "legamart" argument
- Begins with ETML for one or more data marts
- Establishes a common staging area called a Dynamic Data Store (DDS) for ET results, including a common or global metadata repository
- Constructs EDMA before beginning data marts
- EDMA includes enterprise subject areas, and common dimensions, metrics, business rules, sources, all represented in a logically common (but not necessarily physically centralized) metadata repository

Enterprise Data Mart Architecture (EDMA) (Two)

- Uses Dimensional Data Models for Data Marts
- Develops the data warehouse and data marts from the metadata repository, data marts and the DDS using an incremental approach
- Informatica's PowerCenter Tool was the first to implement this architecture. Informatica supports metadata management through monitoring and reporting mechanisms, not through an automated process. Informatica makes some use of Object Technology. Carleton is also close to this capability.

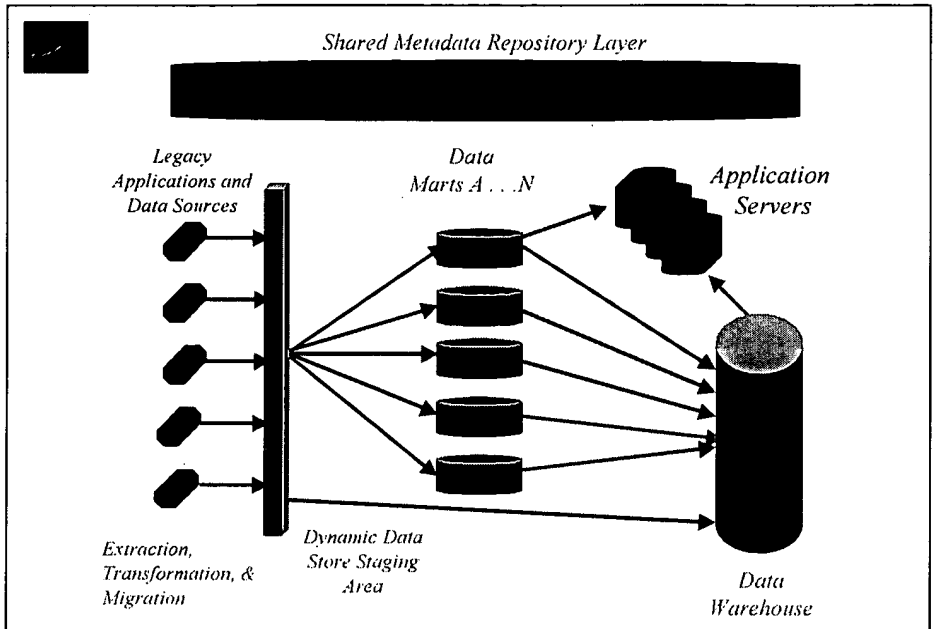


Figure Three -- EDM Architecture

Data Stage/Data Mart Architecture (DS/DMA)

- The same as EDMA with the important exception that no physical enterprise-wide data warehouse is implemented
- Instead, the data warehouse is viewed as the conjunction of the data marts in the context of an EDMA-like metadata repository
- The repository provides a common view of enterprise DSS resources, but not necessarily an enterprise-wide view, because there is no

Data Stage/Data Mart Architecture (DS/DMA) (Two)

- guarantee that the conjunction of data marts will provide access to enterprise-wide global attributes, as would a data warehouse
- Leading tool providers are again Informatica and Carleton, and also Sagent which is advocating DS/DMA in association with Ralph Kimball

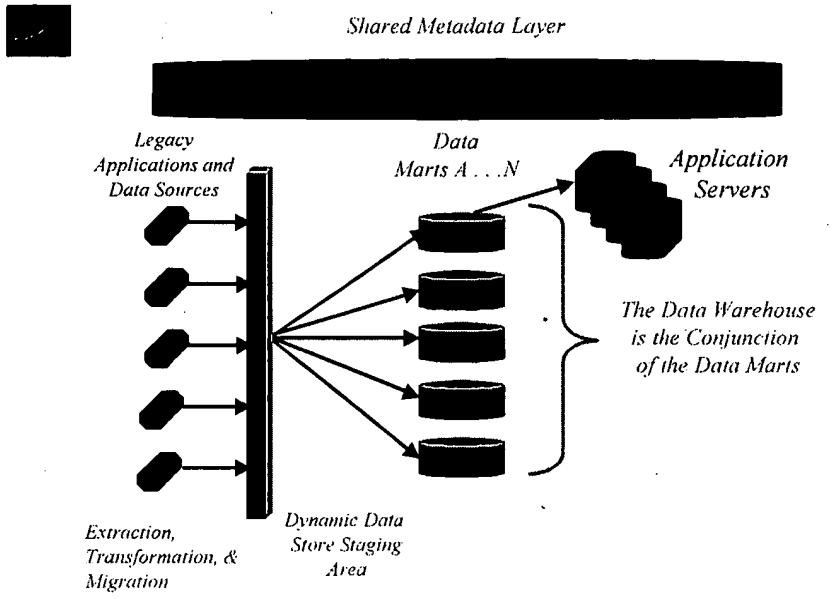


Figure Four -- DS/DM Architecture

Distributed Data Warehouse/Data Mart Architecture (DDW/DMA)

- Again similar to EDMA. Also:
- Provides common view of metadata across the enterprise
- Provides a logical database layer mapping a unified logical data model to physical tables in the various data marts and the data warehouse
- Provides transparent querying of a unified logical database across data marts along with caching and joining services.

Distributed Data Warehouse/Data Mart Architecture (DDW/DMA) (Two)

- Leading tool providers are Informatica, Carleton, Sybase Adaptive Server, and HP (now Platinum) Intelligent Warehouse
- These tools (except IW) are all offered as part of Sybase's Warehouse Studio
- This is the most adaptable of the architectures discussed to this point, but it still reflects the limitations of the relational viewpoint when it comes to handling objects and processes, and it still doesn't support distributed and automated change capture and management

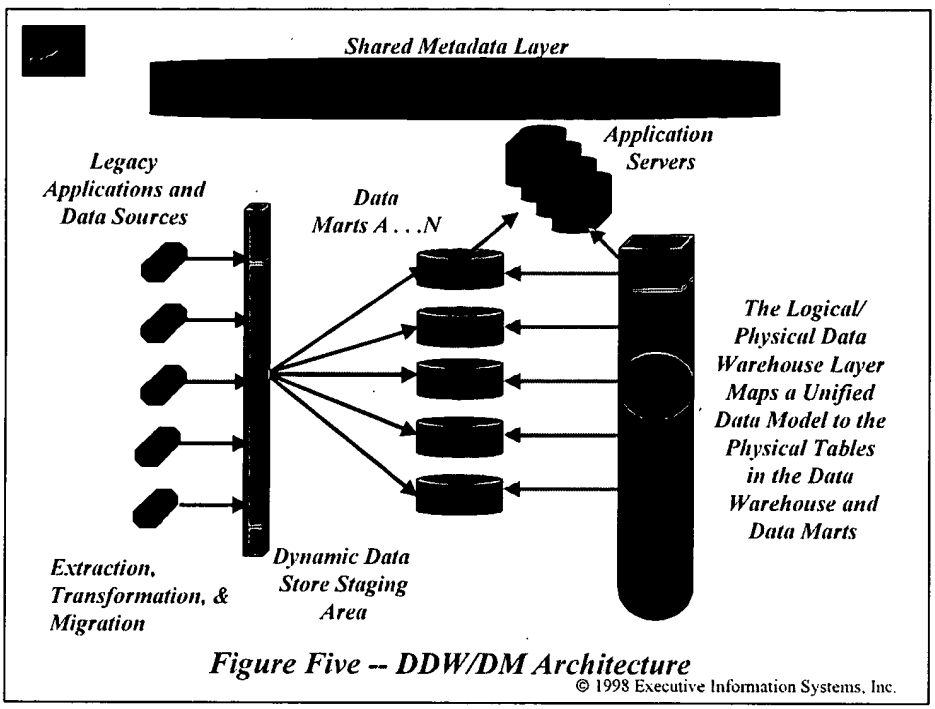


Figure Five -- DDW/DM Architecture

Distributed Knowledge Management Architecture (DKMA)

- An evolving O-O/Component-based architecture
- Top - Down and Bottom-Up architectures may be viewed as two-tier architectures utilizing clients and local or remote databases
- EDMA, DS/DMA, and DDW/DMA may be viewed as adding Middleware and Tuple layers to earlier architectures to provide the capability to manage warehouse systems integration through unified logical views, monitoring, reporting, and intentional DBA maintenance activity. But this

Distributed Knowledge Management Architecture (DKMA) (Two)

- form of management still doesn't provide automatic feedback of changes in one component to others
- DKMA may be viewed as adding an object layer to EDMA or to DDW/DMA to provide integration through automated change capture and management

Distributed Knowledge Management Architecture (DKMA) (Three)

- The object layer contains process distribution services, an in-memory active, object model, and connectivity to a variety of data store and application types. The layer requires an architectural component called an Active Knowledge Manager (AKM).

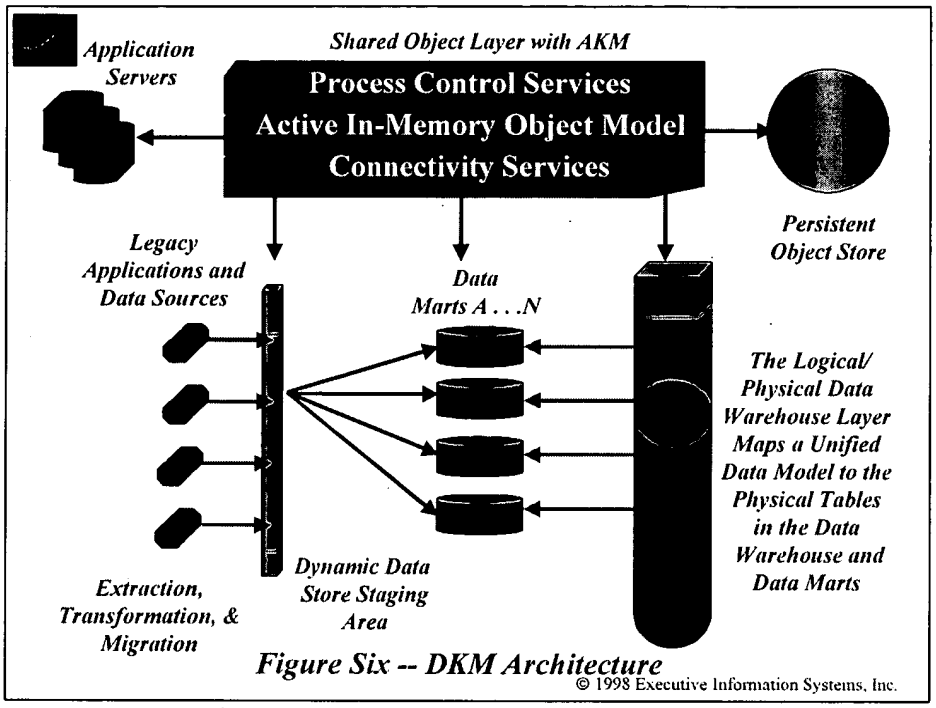


Figure Six -- DKM Architecture

DKM Architecture and the AKM

- An AKM provides Process Control Services, an Object Model of the Distributed Knowledge Management System (DKMS) (the system corresponding to the DKM architecture), and connectivity to all enterprise information, data stores, and applications
  - Process Control Services:
    - In memory proactive object state management and synchronization across distributed objects
    - Component management
    - Workflow management
    - Transactional multithreading

DKM Architecture and the AKM (Two)

- In-memory Active Object Model/Persistent Object Store is characterized by:
  - Event-driven behavior
  - DKMS-wide model with shared representation
  - Declarative business rules
  - Caching along with partial instantiation of objects
  - A Persistent Object Store for the AKM
  - Reflexive Objects
- Connectivity Services should have:
  - Language APIs: C, C++, Java, CORBA, COM
  - Databases: Relational, ODBC, OODBMS, hierarchical, network, flat file, etc.

DKM Architecture and the AKM (Three)

- Wrapper connectivity for application software: custom, CORBA, or COM-based.
- Applications include all the categories mentioned in the earlier discussion of the Dynamic Integration problem, whether these are mainframe, server, or desktop - based.

- The DKM Architecture and the AKM provide the solution to the Dynamic Integration Problem, because only the DKMA among the preceding architectures supports distributed proactive monitoring and management of change in the web of data warehouse, data mart, web information

DKM Architecture and the AKM (Four)

- servers, component transaction servers, data mining servers, ETML servers, other application servers, and front-end applications comprising today's Enterprise DSS/Data Warehousing System.

ODS Variations

- Each of the architectures covered may vary with the addition of an Operational Data Store (ODS)
- According to Inmon: "An ODS is a collection of data containing detailed data for the purpose of satisfying the collective, integrated operational needs of the corporation . . . The ODS is:
  - subject-oriented,
  - integrated,
  - volatile,
  - current-valued,
  - detailed."

ODS Variations (Two)

- The ODS is like a data warehouse in its first two characteristics, but it is like an OLTP system in its last three characteristics. Its purpose is to support operational, tactical decisions
- The workload of an ODS involves four kinds of processing: loading data, updating, access processing, and DSS-style analysis across many records.
- The four types of ODS processing are the source of difficulties in optimizing ODS processing. It is difficult to optimize performance over all four types.

ODS Variations (Three)

- Look at the above architectures in relation to the ODS. It is clear that an architecture that will support both DSS and OLTP-style processing is needed in order to optimally integrate the ODS into the broader data warehousing architecture. In particular,
  - process control services will be very important for the OLTP-style of processing we find in the ODS.
  - Also, distribution of ODS objects across multiple servers will help ODS performance.

ODS Variations (Four)

- Finally, in-memory processing in distributed AKMs can do much to upgrade performance in a distributed ODS.
- **Of course, only one of the above architectures can provide these capabilities for the ODS: the DKM Architecture.**

DKM Architecture and Data Mining

- A key emerging capability in DKMS and data warehousing systems is Knowledge Discovery in Databases (KDD) or Data Mining.
- The key mechanism for KDD is the data mining server.
- Here are some difficulties with current data mining server products:
  - It's difficult to incorporate new data mining algorithms, and therefore keep pace with new developments coming out of the research world;

DKM Architecture and Data Mining (Two)

- Many products require that data must be transported to proprietary data stores before data mining can occur;
- Models produced by the data mining algorithms are not freely available to power users unless they use the data mining tool itself,
- It is difficult to incorporate validation criteria not initially incorporated in the data mining tool into the KDD process,
- There are few "open architecture" commercial data mining tools.

DKM Architecture and Data Mining (Three)

- To solve these problems a product class called An Analytical Data Mining Workbench (ADMW) should be developed.
- The ADMW needs:
  - Easy and convenient encapsulation of new algorithms into object model classes;
  - Capability to mine data from any data source in the enterprise;
  - Incorporation of analytical models into an object model repository;
  - A modifiable validation model,

DKM Architecture and Data Mining (Four)

- Integration of legacy data mining applications with the ADMW.

- An ADMW with these capabilities would meet all of the difficulties specified above.
- The DKM Architecture can help in developing the above because:
  - New algorithms can be encapsulated in objects through the "wrapping" capabilities of the AKM;
  - Data can be brought into the AKM's in-memory object model for data mining without relocating it from its data store ("chunks," partial instantiation),

DKM Architecture and Data Mining (Five)

- Data mining can be performed by executing the analytical models in memory on data chunks and partially instantiated objects;
- Analytical Models produced by an AKM -based application would be placed in an object model repository where they can be accessed by Power Users;
- Customized validity criteria could be added by modifying the validation model in the repository, because the validation model is just another object whose attributes and methods can be modified ;

DKM Architecture and Data Mining (Six)

- Legacy data mining applications could be integrated using AKM connectivity services which would "wrap" them.

- When viewing the above, keep in mind that there is no ADMW with the above capabilities at present. Data Mining is a rapidly growing field, but the market niche represented by the ADMW is empty.
- On the other hand, there are software tools that can be used as a foundation to rapidly develop the ADMW as a facility within the AKM.

DKM Architecture and Software Tools

- To implement DKM Architecture in a DKMS you need the full range of tools now used to create data warehousing systems. In addition though, you need additional tools for the AKM component (including the ADMW facility and the ability to integrate the ODS into the DKMS). These include:
  - An object modeling RAD environment providing extensive process control services and connectivity (e.g. DAMAN's InfoManager, Template Software's Enterprise Integration Template (EIT), Forte, a combination of Ibex's DAWN workflow product along with its Itasca Active Object

DKM Architecture and Software Tools (Two)

- Database, a combination of Rational Rose, Persistence Power-Tier; and Iona's Orbix)
- Technology for constructing software agents to proactively monitor components of the DKMS (e.g. CA Unicenter TNG, ObjectSpace's Voyager, Persistence Power-Tier, DAMAN's InfoManager)
- An OODBMS to serve as a persistent object repository for the AKM component (ObjectStore, Objectivity/DB. Jasmine, Versant, Itasca)

Back-Up Slides

- Distributed Knowledge Management Systems (DKMS
- Why DKMS?
- What is the Knowledge Management System (KMS)?
- The Knowledge Base and Knowledge
- The Knowledge Management Process and Knowledge Management
- Data, Information, Knowledge, and Wisdom
- Organizational Knowledge

Distributed Knowledge Management Systems (DKMS)

- A DKMS is a system that manages the integration of distributed objects into a functioning whole producing, maintaining, and enhancing a business knowledge base.
- A business knowledge base is the set of data, validated models, metamodels, and software used for manipulating these, pertaining to the enterprise, produced either by using a DKMS, or imported from other sources upon creation of a DKMS. A DKMS, in this view, requires a knowledge base to begin operation. But it enhances its own knowledge base with the passage of time because it is a self-correcting system, subject to testing against experience.
- The DKMS must not only manage data, but all of the objects, object models, process models, use case models, object interaction models, and dynamic models, used to process data and to interpret it to produce a business knowledge base. It is because of its role in managing and processing data, objects, and models to produce, enhance, and maintain a knowledge base that the term Distributed Knowledge Management System is so appropriate.

Why DKMS?

- Other reasons for adopting the term DKMS include:
  - business knowledge production and management is what business intelligence is all about;
  - DKMS plays off DBMS, and therefore capitalizes on a familar term while favorably contrasting with it, i.e. knowledge management is clearly better than mere data management;
  - DKMS also highlights the point that data is not knowledge, but only a part of it;
  - "DKMS" is a product/results-oriented name likely to appeal to business decision makers (that is, they get valuable and valid knowledge that they can use to gain control and produce results);

What is the Knowledge Management System (KMS)?

- The Knowledge Management System (KMS) is the on-going, persistent interaction among agents within a system that produces, maintains, and enhances the system's knowledge base. This definition is meant to apply to any intelligent, adaptive system composed of interacting agents.
- An agent is a purposive, self-directed object.
- Knowledge Base will be defined in the next section.
- In saying that a system produces knowledge we are saying that it (a) gathers information and (b) compares conceptual formulations describing and evaluating its experience, with its goals, objectives, expectations or past formulations of descriptions, or evaluations.
- Further, this comparison is conducted with reference to ***validation criteria***. Through use of such criteria, intelligent systems distinguish competing descriptions and evaluations in terms of closeness to the truth, closeness to the legitimate, and closeness to the beautiful.

What is the Knowledge Management System (KMS)? (Two)

- In saying that a system maintains knowledge we are saying that it continues to evaluate its knowledge base against new information by subjecting the knowledge base to continuous testing against its validation criteria. We are also saying that to maintain its knowledge, a more complex system must ensure both the continued dissemination of its currently validated knowledge base, and continued socialization of intelligent agents in the use and content of its knowledge base.

- Finally, in saying that a system enhances its knowledge base, we are saying that it adds new propositions and new models to its knowledge base, and also simplifies and increases the explanatory and predictive power of its older propositions and models. That is, one of the functions of the KMS is to provide for the growth of knowledge.

The Knowledge Base and Knowledge

- A system's knowledge base is: the set of remembered data; validated propositions and models (along with metadata related to their testing); refuted propositions and models (along with metadata related to their refutation); metamodels; and (perhaps, if the system produces such an artifact) software used for manipulating these, pertaining to the system and produced by it.
- A knowledge management system, in this view, requires a knowledge base to begin operation. But it enhances its own knowledge base with the passage of time because it is a self-correcting system, and subjects its knowledge base to testing against experience.
- Finally, the emphasis on a system's knowledge base, rather than its knowledge, recognizes that an identification of knowledge as individual conceptions, propositions, or models is inconsistent with the reality that acceptance of a piece of information into a system's body of knowledge is dependent on the background knowledge already within the knowledge base. This background knowledge is used to filter and interpret the information being evaluated.

The Knowledge Base and Knowledge (Two)

- This definition of knowledge base contrasts with a popular definition of knowledge as "justified, true belief." The definition does agree with the necessity of justification as a necessary condition for knowledge; but it insists that justification be specific to the validation criteria used by a system to evaluate its descriptions and evaluations. The definition also agrees that knowledge is a particular kind of belief, provided that belief extends beyond cognition alone, to evaluation.

- The biggest discrepancy with the popular definition is in not requiring that justified beliefs be "true." Truth can be used as a regulating ideal by a system producing descriptive knowledge. "Right" can be used as a regulating ideal by a system producing evaluative or normative knowledge. But the system in question can never say for sure that a proposition or a model within its knowledge base is "true," or "right;" but only that it has survived refutation by experience better than its competitors, and therefore that the system "believes" it is true or right.

The Knowledge Base and Knowledge (Three)

- So instead of knowledge as "true, justified belief," the position taken here is that knowledge equals justified belief that some conceptual formulation, fact, or evaluation, is true or right as the case may be.
- In a very real sense, a system's knowledge is ***the analytical network of propositions and models constituting the knowledge base.*** It is therefore, just for convenience, that one may refer to a particular proposition or model as something a system "knows," because it knows that "something," only if one assumes that numerous unspecified background propositions and models are also known by it.

The Knowledge Management Process and Knowledge Management

- The Knowledge Management Process (KMP) is an on-going persistent interaction among human-based agents who aim at integrating all of the various agents, components, and activities of the knowledge management system into a planned, directed process producing, maintaining and enhancing the knowledge base of the KMS.

- Knowledge Management is the human activity within the KMP aimed at creating and maintaining this integration, and its associated planned, directed process.

The Knowledge Management Process and Knowledge Management (Two)

- A good way to look at the human activity called knowledge management is through the concept of the Use Case. In a use case a human-based agent, within the KMS, called an actor, participates in the KMP to get an outcome from the KMS that has value for the actor. The KMP can be represented as a set of Business Process Use Cases each classified within one of four business sub-process categories: planning, acting, monitoring, and evaluating. *A way of decomposing knowledge management activity then, is in terms of the use cases that constitute it.*
- The set of all use cases aimed at creating and maintaining the integrated, planned, directed process producing, enhancing and maintaining the KMS knowledge base, is an alternative characterization of knowledge management. The set of these use cases represents all of the organizational knowledge management activity of the actors making use of the KMS through the KMP. In other words, the set of use cases is what we mean by knowledge management in a human system.

Data, Information, Knowledge, and Wisdom

- What is the difference between data, information, knowledge, and wisdom?
- To begin with, organizational data, information, knowledge, and wisdom, all emerge from the social process of an organization, and are not private. In defining them, we are not trying to formulate definitions that will elucidate the nature of personal data, information, knowledge, or wisdom. Instead, to use a word that used to be more popular in discourse than it is at present, we are trying to specify intersubjective constructs and to provide metrics for them.
- A datum is the value of an observable, measurable or calculable attribute. Data is more than one such attribute value. Is a datum (or is data) information? Yes, information is provided by a datum, or by data, but only because data is always specified in some conceptual context. At a minimum, the context must include the class to which the attribute belongs, the object which is a member of that class, some ideas about object operations or behavior, and relationships to other objects and classes.

Data, Information, Knowledge, and Wisdom (Two)

- Data alone and in the abstract therefore, does not provide information. Rather, information, in general terms, is data plus conceptual commitments and interpretations. Information is data extracted, filtered or formatted in some way (but keep in mind that data is always extracted filtered, or formatted in some way).
- Knowledge is a subset of information. But it is a subset that has been extracted, filtered, or formatted in a very special way. More specifically, the information we call knowledge is information that has been subjected to, and passed tests of validation. Common sense knowledge is information that has been validated by common sense experience. Scientific knowledge is information (hypotheses and theories) validated by the rules and tests applied to it by some scientific community.
- Wisdom, lastly, has a more active component than data, information, or knowledge. It is the application of knowledge expressed in principles to arrive at prudent, sagacious decisions about conflict situations.

Organizational Knowledge

- Organizational knowledge in terms of this framework is information validated by the rules and tests of the organization seeking knowledge. The quality of its knowledge then, will be largely dependent on the tendency of its validation rules and tests to produce knowledge that improves organizational performance (the organization's version of objective knowledge).

- From the viewpoint of the definition given of organizational knowledge, what is an organization doing when it validates information to produce knowledge? The validation process is an essential aspect of the broader organizational learning process, and that validation is a form of learning. So, though knowledge is a product and not a process derived from learning, knowledge validation (validation of information to admit it into the knowledge base) is certainly closely tied to learning, and depending on the definition of organizational learning, may be viewed as derived from it.

Issues in Developing Very Large Data Warehouses

Lyman Do Pamela Drew Wei Jin Vish Jumani David Van Rossum

Applied Research and Technology
Shared Services Group
The Boeing Company
P.O. Box 3707, M/S 7L-70, Seattle, WA 98124-2207, USA
Email: {Lyman.S.Do, Pamela.A.Drew, Wei.Jin, Vish.Jumani, David.A.VanRossum}@boeing.com

Abstract

The size of The Boeing Company posts some stringent requirements on data warehouse design and implementation. We summarize four interesting and challenging issues in developing very large scale data warehouses, namely failure recovery, incremental update maintenance, cost model for schema design and query optimization, and metadata definition and management. For each issue, we give the reasons we think it is important but not well-addressed in research literature and commercial products, and our current research to solve it.

1 Introduction

Several data warehouse development projects are being pursued in Boeing with sizes ranging from hundreds of megabytes to terabytes. Some projects are aimed at providing sophisticated decision support and some are designed to re-distribute the workload of OLTP systems. In the latter, some large read-only queries will be re-directed to a data warehouse in order to relieve the heavy workload of our OLTP systems. This paper raises four issues that are challenging to the development of large-scale data warehouses.

The size of Boeing posts stringent requirements on data warehouse design. The largest data warehouse project will have an initial size of two to three terabytes and will operate on a 24x7 basis. Each airplane typically has over one million parts and the information of all parts and the airplane configuration will be kept in the warehouse until the airplane itself is retired. This means that the data lifecycle could be as long as 70+ years, instead of 5 to 10 years in most companies. This also implies that the data warehouse will continue to grow for 70 years. In addition to size, a long data lifecycle imposes a heavy requirement on the evolution of the data warehouse which has to be flexible enough to access 70 year old data. Moreover, since Boeing is a global company, there is virtually no "nighttime" for data warehouse refresh and maintenance. The refresh window is small and may not be extended. Any failure during refresh may miss the refresh window which delays business decisions or processes. The size of the data warehouse, the length of the data lifecycle, the flexibility to access data, and the strict demands on system availability establish requirements that challenge the most sophisticated technical solutions in data warehouse designs and implementations today.

In the following sections, we present four issues in turn and discuss the research directions that we are working toward.

2 Failure Recovery

Failure can happen anywhere, from populating the data warehouse, to refreshing multi-dimensional databases, to processing end-user queries; each task is long and resource intensive, and each task is costly to roll-back and restart. The situation becomes worse if the refresh window is small and can not be missed. This issue is not well addressed because failure recovery discussions typically only encompasses writing data into persistent storage but in a data warehouse environment, the most costly failures happen during computation such as data cleansing, aggregation/roll-

Proceedings of the 24th VLDB Conference
New York, USA, 1998

up, indexing, etc.

Typical transactional roll-back and restart failure recovery is not applicable in the above scenario because 1) it is too expensive (time and resources) to rollback a long-lived task, 2) the recovery log may not be available or too expensive to store due to the large volume of updates, and 3) in a global enterprise, the refresh window of a data warehouse is too small to allow rollback and re-start.

There are two types of failure that we have been looking into: (1) database/data warehouse update failure that happens during the data population and refresh from data sources to data warehouse and (2) computation failure during end-user query processing. The former is similar to a typical database failure except that it can happen even before writing records into the database during data cleansing and scrubbing, or it can happen during index or metadata updates.

The second type of failure happens during query processing. A typical decision support query may scan through multiple tables and require intensive computation to prepare summary information. Failure in any step during the computation requires re-work that is expensive in terms of query response time for mission critical queries, and computational resources. Also, re-work may eventually delay the warehouse refresh window. The influence of re-work and delay is further amplified since the data warehouse supports a large user population.

We are looking into the technology of incremental checkpointing to provide forward recovery. Ideally, all long-lived tasks, such as data cleansing, warehouse population and refresh, data summarization, indexing, roll-up, and query processing, should incrementally write checkpoints to a persistent storage. In case of failure, the system only needs to have partial roll-back to previous checkpoint and re-start. The concept is simple but technically challenging. Incremental checkpointing requires modularization of those long-lived tasks by analyzing and decomposing a long-lived task into a pipeline of sub-tasks each of which is loosely coupled with the others. Incremental checkpointing is performed between sub-tasks. Sub-tasks should be loosely coupled so that in case of failure, the system can roll-back to a previous checkpoint for each sub-task and re-start the pipeline. Another issue related to the incremental checkpointing is the need of an efficient and generic logging facility that provides persistent logging for checkpoints of different tasks.

3 Incremental Update Maintenance

In addition to the differential relation [OV91] approach of incremental update, we need a mechanism to support data sources that do not export differential relations. A differential relation captures the before image and the after image of all tuples that each operation affects. Most research work on data warehouse update focuses on the problem that "given a differential relation, how do we refresh the data warehouse efficiently." These works differentiate each other in terms of different data warehouse capabilities, such as convergent warehouse consistency [ZGHW95,ZHW96], replication of some source relations [QW97], versioning [QGMW96], etc. They are all based on the same assumption that differential relations are available. Likewise, commercial products either suggest refreshing the data warehouse from scratch (the snapshot approach) if the refresh window is large enough or support incremental update using differential relations.

Such an assumption may not be valid for the reason that some vital production systems do not export differential relations. Even if we violate the local autonomy by modifying the application code to extract differential relations from each database update, it is expensive to extract the before and after image of a SQL statement. To do this, the system needs to run a modified SQL statement with the same FROM and WHERE clauses as the original update operation before the execution of that update operation, then execute the update operation, then run the modified SQL statement again to collect the after image. Triggers could be helpful if the trigger can be fired before and after each execution (for INSERT, DELETE, and UPDATE) and if the triggers are tightly coupled with the update operation, i.e., being executed in a single atomic transaction. To further complicate the situation, some production systems use object wrappers to encapsulate relational schema or complete transactions and some commercial applications make it extremely difficult to interpret data storage structures, not to mention the feasibility of implementing triggers on them.

We have been working on incremental data warehouse update maintenance by capturing the operation descriptions at the sources. To differentiate, we refer to the differential relation as *value-delta* and the operation description as *operation-delta (Op-delta)*. We are motivated by the fact that Op-delta could be extracted from database log and it is less expensive, in terms of storage, communication, and computation overhead, for programs to export Op-delta instead of the value-delta. For example, the statement: *"UPDATE status='revised' from parts where last_modified_date > 1/1/98"* may generate a value-delta in the size of ten thousand records but the SQL statement itself is already an Op-delta in the size of 70 bytes. We have identified sufficient conditions that Op-delta alone is enough to refresh the data warehouse (i.e., self-maintainability with respect to Op-delta),

and for some cases, a hybrid between value-delta (the before image portion only) and the Op-delta is necessary to refresh the data warehouse. In both cases, the data warehouse does not need to refer back to the source during the refresh.

Another advantage of Op-delta is allowing the data warehouse to refresh concurrently with end-user query processing. The data warehouse treats a refresh as a series of execution of Op-delta. This implies that there will be virtually no downtime for data warehouse refresh, i.e., minimize or eventually eliminate the update window. To achieve this, we are currently working on the definition of data warehouse consistency in terms of concurrent refresh and concurrency protocol(s) for the data warehouse refresh.

4 Cost Model for Schema Design and Query Optimization

We need a cost model to analyze the cost and benefit of designing data warehouse schema. At large, the cost model should help in selection of data model among relational schema, star schema, and multi-dimensional database. At a finer granularity, the cost model should help in determining the dimensions of a multidimensional cube or in a star schema.

During the design of a data warehouse, an intuitive requirement is to maximize query performance. The resulting products are the star schema and multidimensional databases that pre-compute a sub-set of the most frequently asked queries (or asked by the most important person) and materialize the result. It is obvious that the query response time is tremendously improved but it is less obvious (or promoted) that a larger maintenance window is implied. Our traditional database training tells us that materialized views can improve query performance if we can manage to update the views consistently, i.e., we are trading data warehouse update maintenance cost for better query response time. But questions such as where is the balance point between improved query response time and the minimal maintenance window arise. There is no cost model to provide guidelines on how much information we should pre-compute and materialize, what kind of queries can benefit most from a materialized view, what is the cost to maintain a star schema or the equivalent multidimensional cube, etc.

In the Boeing Company, each airplane has potentially one million attributes to describe it. It is impossible to develop a multidimensional cube that has one million dimensions. An intuitive question will be "which attribute should we include in the limited multidimensional cube?" What is the benefit to introduce one more dimension and what will be the cost to maintain it? Again, we are looking for a cost model that can analyze the cost and benefit of bringing additional dimension into a multidimensional cube. The argument remains valid for a star schema. Assuming that the fact table is populated from tables in a normalized relation source, adding one more dimension table to the star schema means adding one more table to the join query that prepares the fact table. The cost to populate the fact table will then increase exponentially.

Last but not least, if a data warehouse cannot answer a query (determining whether a query is answerable is yet another issue), the query will then be reformulated and submitted to operational databases. What are the criteria that a query or a particular set of queries should be supported by the data warehouse? What is the cost of materializing additional relations/multidimensional cubes in the data warehouse in order to reduce the number of queries that have to be submitted to operational databases? Again, we need a cost model to analyze the balance among the cost of query processing at operational database, the cost of data warehouse update maintenance, and the benefit of supporting that query by the data warehouse.

5 Metadata Definition and Management

An orthogonal issue is the metadata definition and management. Example metadata includes: information about the data source such as the cost model of its query processing, whether value-delta is supported, whether it is possible to extract the Op-delta; information about the data such as source schema, data warehouse schema, and their mappings, update frequency and the average size of update; information about queries such as the cost to process a particular query at source, cost to process the same query if a multidimensional cube supports it, the frequency of the query, the importance (priority) of the query; information about the data warehouse such as the schema definition, subject area of each multidimensional cube/fact table, maintenance window, cost of maintenance, etc.

Metadata begins to accumulate at the very beginning of a data warehouse development project, either physically or electronically, and it grows during the development and beyond. We are particularly interested in the metadata that is 1) involved in the design/development decisions, 2) referred to during normal operations (data population, cleansing, refresh, etc.) of a data warehouse, and (3) referred to during end-user query processing. The first type of metadata includes the business model of data stored in operational databases and the cost model described

in Section 4. This type of metadata is used to identify what information should be included in the data warehouse and from where to populate and refresh the identified information. The second type of metadata is used to maintain the data warehouse by identifying the methods to refresh the data warehouse, and for each part (multidimensional cubes, fact tables, etc) of the data warehouse, how to perform the refresh, how frequent the refresh should be performed, and when to archive the historical information. The third type of metadata helps users to identify information or subjects that are available in the data warehouse. It helps users to determine if a query is supported by the data warehouse. In an extreme, metadata helps the query processing system automatically route a query to the data warehouse or to the operational databases where the query can be executed, allowing a better response time and lower cost. End-user understanding of the data is generally not based on a relational type model, but a non-computing, business model, thus requiring a mapping of a business model (and possibly the business processes) to the physical data model used by the data warehouse. Due to the magnitude of the metadata, an online access with a business interpretation must be available.

After metadata is defined and the sources are identified, we need to manage changes to the metadata. Change management should capture changes at heterogeneous and distributed data sources and propagate the changes to a metadatabase. Besides, we need tools to analyze the changes and evolve the data warehouse. For example, changing the schema definition at a data source will change the metadata of that schema at the data warehouse and a mapping function between source to data warehouse tables/cubes. The change may also affect the self-maintainability of data warehouse tables/cubes, which in turn supports a more effective refresh mechanism. Likewise, changes of queries, their frequency, and priority may also trigger similar evolution at the data warehouse.

We are working on the definition of metadata, the schema definition in the "metadatabase", the software components in the "metadatabase" that supports queries on the metadata and that supports continuous update (change management) of the metadata.

6 Summary

Like most global enterprises, Boeing is looking into the data warehousing solutions to improve end-user query performance, to re-distribute some long-lived read-only queries from our overloaded OLTP systems, and to support a new-generation of decision support systems. We presented four interesting and challenging large-scale data warehouse development issues and we are actively working toward the solutions of them.

References

Our references include the research work at AT&T Labs, Bell Labs, Stanford University, and various commercial data warehouse products. For brevity, we only include some of the references here. We direct interested readers to Alberto Mendelzon's "Data Warehousing and OLAP: A Research-Oriented Bibliography" web page at University of Toronto (http:\\ www.cs.toronto.edu/~mendel/dwbib.html).

[OV91] M.T. Özsu and P. Valduriez. *Principles of Distributed Database Systems*. Prentice-Hall, 1991.

[QGMW96] D. Quass, A. Gupta, I.S. Mumick, and J. Widom. Making views self-maintainable for data warehousing. In *Proceedings of the Sixth International Conference on Parallel and Distributed Information Systems*, Miami Beach, Florida, 1996.

[QW97] D. Quass and J. Widom. On-line warehouse view maintenance. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 405-416, Tucson, Arizona, May 1997.

[ZGHW95] Y. Zhuge, H. Garcia-Molina, J. Hammer, and J. Widom. View maintenance in a warehousing environment. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 316-327, San Jose, California, May 1995.

[ZGW96] Y. Zhuge, H. Garcia-Molina, and J.L. Wiener. The Strobe algorithms for multi-source warehouse consistency. In *Proceedings of Conference on Parallel and Distirbuted Information Systems*, Miami Beach, Florida, 1996.